AD-A100 652 GEORGIA INST OF TECH ATLANTA SCHOOL OF INFORMATION A--ETC F/G 9/2
OPERATIONAL SURVIVABILITY IN GRACEFULLY DEGRADING DISTRIBUTED P--ETC(U)
DEC 80 E W MARTIN N00014-79-C-0231
UNCLASSIFIED GIT-ICS-81/04 NL

1 OF 1
AD A
100652

END
DATE
FILMED
7-81
DTIC

LEVEL II

AD ___________

AD A100652

TECHNICAL REPORT
GIT–ICS–81/04

OPERATIONAL SURVIVABILITY IN GRACEFULLY DEGRADING DISTRIBUTED PROCESSING SYSTEMS

By
Edith Waisbrot Martin

Prepared for
OFFICE OF NAVAL RESEARCH
800 N. QUINCY STREET
ARLINGTON, VIRGINIA 22217

Under
Contract No. N00014–79–C–0231
GIT Project No. G36–636

December, 1980

GEORGIA INSTITUTE OF TECHNOLOGY
SCHOOL OF INFORMATION AND COMPUTER SCIENCE
ATLANTA, GEORGIA 30332

THE RESEARCH PROGRAM IN
FULLY DISTRIBUTED PROCESSING SYSTEMS

OPERATIONAL SURVIVABILITY
IN
GRACEFULLY DEGRADING
DISTRIBUTED PROCESSING SYSTEMS

TECHNICAL REPORT
GIT-ICS-81/04

Edith Waisbrot Martin

December, 1980

Office of Naval Research
800 N. Quincy Street
Arlington, Virginia 22217

Contract No. N00014-79-C-0231
GIT Project No. G36-636

The Georgia Tech Research Program in
Fully Distributed Processing Systems
School of Information and Computer Science
Georgia Institute of Technology
Atlanta, Georgia 30332

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

14

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER
GIT-ICS-81/04

2. GOVT ACCESSION NO.
AD-A100 652

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)
6 Operational Survivability in Gracefully Degrading Distributed Processing Systems. 9

5. TYPE OF REPORT & PERIOD COVERED
Technical Report

6. PERFORMING ORG. REPORT NUMBER
GIT-ICS-81/04

7. AUTHOR(s)
10 Waisbrot
Dr. Edith [redacted]/Martin 15

8. CONTRACT OR GRANT NUMBER(s)
N00014-79-C-0231

9. PERFORMING ORGANIZATION NAME AND ADDRESS
School of Information and Computer Science
Georgia Institute of Technology
Atlanta, Georgia 30332

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
C-

11. CONTROLLING OFFICE NAME AND ADDRESS
Office of Naval Research
800 N. Quincy Street
Arlington, Virginia 22217

12. REPORT DATE
11 December 1980

13. NUMBER OF PAGES
155 + xi

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)
12 | 86

15. SECURITY CLASS. (of this report)
Unclassified

15a. DECLASSIFICATION DOWNGRADING SCHEDULE
N/A

16. DISTRIBUTION STATEMENT (of this Report)

The United States Government is authorized to reproduce and distribute reprints for government purposes

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

The findings in this report are not to be construed as an official Department of the Navy position unless so designated by other authorized documents.

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Distributed Data Processing
Survivability
Performance Evaluation
Data Analysis
Multiple Regression

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Under this contract a simulator was designed and developed which will model possible distributed system network topologies, distributed system application topologies and their effect on application system performance as the configuration of the distributed system network is continuously and arbitrarily reduced. The objective of the model is to aid in development of a measure of survivability which can subsequently be used to evaluate and compare alternative distributed system designs for specific battlefield applications.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

Unclassified

410044

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

SUMMARY

To date the concept of survivability as it pertains to distributed processing systems has been an intuitive one. The objective of this research is to present this concept quantitatively. Toward this end a number of hypotheses are presented, namely, that survivability must be measured in a nontrivial or indirect manner; that survivability is a function of a number of attributes, all of which are necessary to adequately explain or predict survivability; that the attributes which describe survivability are large in number and complex in interaction; and that because of these characteristics traditional performance, survivability and reliability measures are inadequate. This research proposes to demonstrate the applicability of standard experimental design and regression analysis techniques to the field of computer science in general and modeling of distributed systems in specific.

To test these hypotheses a computer model which supports the simulation of distributed processing systems for the purpose of evaluation and experimentation was constructed. This model, called SURSIM, models distributed system networks, application systems and several distribution/redistribution approaches and the effect of these on application system performance as the configuration of the distributed system network is continuously and arbitrarily reduced. In specific, the simulator represents and manipulates those attributes believed to be important in evaluating the survivability of distributed processing systems which must operate in real-time environments such as battle-

field situations. The controlled factors include; distributed system topology, network size (i.e., number of nodes); node processing, memory and communications capacity; applications system size, connectivity and interaction requirements; distribution strategies and extent of distributed system degradation.

SURSIM comprises modeling of the application system, distributed processing system, assignment of the application system to the distributed processing system, and mutation of the distributed system with subsequent reconfiguration of the application system. The capability to analyze application system performance based on application system requirements and distributed system capabilities is provided. In addition, SURSIM has the ability to implement degradation procedures which reduce software application system requirements to accommodate degraded distributed system capabilities. Considerable input and operational data logging is performed as the simulator is exercised. Tables and matrices describing the distributed and application systems being examined are output. Experimental results are generated in tabular and machine readable form to facilitate manual and computerized analysis.

A 2^{K-P} Fractional Factorial experiment was conducted using the simulator as an experimental tool. One hundred and twenty-eight experimental cases were run in which 11 different factors were manipulated and 46 derived measures monitored. The results of the 128 experiment runs and the subsequent 300,000 subcases were examined using a number of statistical techniques. Several approaches to regression modeling such as stepwise and all possible subset regression were used

to build explanatory models from the results collected. Ten of these models proved to serve well in an explanatory capacity, consequently, data splitting was employed to assess the value of these models when used as predictors. It was determined that three models which function well in an explanatory role also serve well in predicting survivability and performance.

Thirty-two candidate regressors are used in identifying the 10 best subset models. The coefficiients of these regressors are approximately equivalent in sign and magnitude across models. All variables remain proportional with the introduction and removal of other variables, thereby demonstrating extreme stability. The explanatory adequacy of models built using these variables is in all instances in excess of .8 which is very acceptable for a factor screening experiment. The adequacy in prediction of these models ranges between -.39 and +.71 with some models predicting very well and other predicting very poorly. By constructing satisfactory explanatory and predictive models, this research demonstrates that the concept of operational survivability and performance as proposed can be expressed quantitatively. Further, it is shown that major factors include the distributed system network, application system and distribution policy as initially hypothesized. Nine factors are found in all models. These are number of nodes in the distributed system, distributed system connectivity, module memory requirements, module to module interaction frequency, distribution policy, percent nodes lost, initial assignment results, available processing capacity at the end of the subcase and the interaction of all application related variables.

The research conducted here identifies the variables important to operational survivability and to some extent tells how large changes in these important variables affect the response. Future experimentation which provides either a large number of factor levels or finer granularity in possible variable values should permit greater resolution in the simulator results and their subsequent application. The results presented in this dissertation demonstrate the applicability of traditional experimentation and regression analysis in the field of computer science as well as the feasibility of measurements which can serve as measurements for distributed systems. The models developed represent a promising initial step in the quantification of operational survivability as it applies to gracefully degrading distributed processing systems.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF ILLUSTRATIONS

CHAPTER I

INTRODUCTION

Overview

The effectiveness of any distributed system design must be viewed against a backdrop of predetermined weights and priorities. To many users, the main benefit to be derived from the distributed approach to application system processing is increased capability to satisfy application system requirements despite the loss of a portion of the distributed system resources. The extent of that capability is herein termed "survivability." Inherent in the concept of survivability is that of "graceful degradation." Gracefully degrading systems are those which attempt to provide a high quality of service by reconfiguring the system or network or by reallocating resources when a fault is detected. This term is used to imply that performance may decrease with successive failures but it may not be catastrophically effected.

This research investigates the concept of survivability in *gracefully degrading* systems. It examines distributed system resources, processing nodes and associated links, which can be lost before a given application system required to execute on that distributed system must function in a degraded mode or experience failure. Whereas the determination of survivability has thus far been primarily judgemental based on a spectrum of performance variables, it is the intent of this research to express this concept quantitatively. Toward that end the applicability of standard experimental design methods and

regression analysis techniques to issues in performance evaluation are demonstrated.

Evaluation of survivability is of importance, for example, to the U.S. Army. Modern warfare has made automation on the battlefield essential to provide the commander with timely information on which to base his decisions and for weapon system and equipment control. Automation is required to enhance the speed, accuracy and dependability of battlefield systems that perform the functions of command, control, communications, intelligence, air defense, weapons control, surveillance, electronic warfare, sensor control, field artillery, navigation, logistics, and administration. Foremost among the goals for battlefield automation are operational effectiveness and continuity of operations. The ability to meet these goals is determined by 1.) the quality of the automation hardware and software components, 2.) the compatability or interchangeability of these components, and 3.) the capability of the components to cooperate together to accomplish assigned tasks.

As an example, the Military Computer Family (MCF) program proposes to address each of these issues. The MCF program addresses quality of hardware and software components by utilizing the state-of-the-art instruction set architecture (ISA) and high order language (HOL) which appear to best fit the projected needs of Army automation systems. These were selected with extreme emphasis on potential reliability, performance and maintainability attributes. The MCF hardware and software members were chosen for their ability to meet the widest possible spectrum of Army automation system requirements and

thereby provide the foundation or standard for such systems. Via standardization the issue of interchangeability is accommodated. This research addresses the third segment of the operational effectiveness/continuity of operations duo, that is the capability of the MCF members to function together as a distributed system serving the requirements of specific Army application systems in battlefield situations.

Distributed System Design Considerations

In distributed systems the concern is for systems composed of many processing and memory components working together to serve a common application. For the most part designs are desired which provide decentralized control of the system, that is the controller does not reside in a single processor. Distributing application system tasks over a number of processing components can result in greater computing speed and capacity than is possible with a single processor of the same approximate cost. This benefit is achieved by customizing the selection and configuration of components to best match application system requirements, and thereby minimizing inefficient use of computer system resources. Distributed systems are believed to be inherently less costly to modify or upgrade because single, relatively small, components can be added or replaced rather than whole systems. Decentralization of resources and application system processing can yield additional benefits with respect to reliability and fault tolerance in that distribution of resources and activities can be so arranged such that the likelihood of single-point failures is reduced. There are numerous ways in which utilization of distributed systems can be advantageous. Currently, many advantages are obtained through

distribution over processing components which are physically close, i.e., within a one mile radius of one another. Distribution at more geographically separated locations entails additional complexity due to transmission delays and increased susceptibility of the distributed system to failures associated with communication losses or noise. Many applications, however, such as military real-time systems, require interaction with geographically dispersed system components. This dispersion may be for reasons inherent in the application or for purposes of system survivability or reduced vulnerability to loss of a portion of the distributed processing system locations. In this research the proximity of the processing components will be important as well as the qualification that the processing components be operating together to serve a single application.

Designing distributed processing systems to-date is not a well understood practice. This is true in part because the distributed system concept is still somewhat new and because it introduces additional variables and complexities into the design process. For example, distributed systems contain concurrent processes which must share resources and data without the benefit of centralized control, data and application system processes are distributed and possibly replicated at multiple locations, and communication management and protocols are often cumbersome and complex. Each of these design issues in the area of distributed processing systems appears to be more complex than its centralized counterpart. Thus far, no general design approach exists which adequately addresses the extremely complex and varied goals of distributed systems. Among the unresolved design

issues are database distribution and management, distributed control, task distribution, fault tolerance, and performance prediction.

CHAPTER II

BACKGROUND

There has been considerable research over the past twenty years in the area of fault tolerant computing. Initially the focus of that research was on the hardware of single processor systems. Fault tolerant computer investigations centered on models of ultra reliable systems having long mission time requirements such as those used in space exploration (19). These were typically uniprocessor systems with requirements for extremely large mean-time-to-first-failure (MTFF). Loss of a processing node was tantamount to catastrophic failure of the function or system served. The software support component of such systems was small and uncomplicated usually consisting of some minimum executive support required to execute the application software.

Later applications with large continuous processing demands presented a need for computer systems with high availability (3,4). The increased throughput and reliability required to support these applications was achieved by the introduction of a special class of multiple processor systems. These were repairable computer systems which embodied the concepts of redundancy and standby sparing (16,4). The important criterion of processor reliability shifted from mean-time-to-first-failure to mean-time-between-failures (MTBF). Support software was more complex than before. Software design had to address issues such as placement of software tasks and monitoring of hardware system components to detect failures and institute recovery

procedures. No matter what hardware backup scheme was employed, the reliability of the system became much more noticeably dependent upon the operation of the executive software. Software control for the most part was either centralized often realizing an underlying master slave relationship or functionally dispersed. As a consequence of this relationship, systems were still extremely vulnerable to single point failure.

Vulnerability to single point failure in part motivated development of distributed hardware and software systems which incorporated distributed control. Prior reliability and fault tolerance concepts laid the foundation for a new system reliability approach which attempts to provide a high quality of service by reallocating resources, i.e. reassigning tasks, or by reconfiguring the system or network, i.e. changing physical interconnection or routing algorithm, when a fault is detected. Such systems are termed "gracefully degrading" systems. This term is used to imply that performance may decrease with successive failures but it may not be catastrophically effected. Techniques for graceful degradation are particularly useful when applied to loosely coupled processor systems such as networks or fully distributed systems, that is, systems the components of which have high potential for autonomous operation. Many Command, Control and Communication, and tactical systems, for example, have significant impact if they experience catastrophic failure. The consequence of failure justifies the additional expense of hardware and software needed to allow military systems to withstand partial system failures.

Graceful degradation techniques are not in wide use currently. In part this is true because there is still a great lack of knowledge of the software organization required to implement graceful degradation. Also, there is a lack of adequate analytical models with which to evaluate such systems. Some research (2,17,22,12) in the area of gracefully degrading systems has been conducted for multiple processor (tightly coupled) systems. These models and the resulting measures, while invaluable to our confidence in tightly coupled multiprocessor systems or loosely coupled systems in which all processors are performing similar functions, prove less meaningful when applied to a large portion of loosely coupled distributed systems such as those used for defense. The main reason is that these analytical models lack consideration for hardware and software topology factors.

Some recent analytical research which addresses the issue of survivability in gracefully degrading systems attempts to accommodate hardware topology and software allocation features of those systems (14,19). One effort by Merwin and Mirhakak proposes that distributed systems are made up of hardware networks and software systems (19). The networks comprise nodes and links. The software system is made up of programs and data. Several failure modes are described. Failure can be caused by loss of a link between a program and its data, loss of the node on which the data resides or loss of the node on which the program is to execute. Failure probabilities are assigned to each node and link. Survivability is determined as the number of programs that remain operational after some combination of nodes and links have failed. A survivability criterion is developed which is based on the

probability of occurrence of any subarchitecture of a given distributed network and the expected number of programs operable for each subarchitecture. A subarchitecture here is defined as any combination of nodes and links which is a subset of the original network configuration. Survivability is expressed as a function of 1.) an initial network architecture, 2.) a given data set distribution and 3.) data set access requirements. The major departure of this work from preceding research is the inclusion of software distribution into the computation of survivability. This model like other analytical models faces several difficulties.

The first problem is computational. In (19) presented above, a number of additions and enhancements to their analytical model are proposed such as weighting of programs and nodes and placing constraints on the data set distribution. However, since the algorithm they use for computing survivability already demonstrates exponential computational growth and complexity as the number of nodes and communications links increase, additional criteria might only serve to exacerbate the present computation problem.

The second problem for the analytical approach is validation. Like other wholly analytical models of distributed systems, the model proposed by Merwin and Mirhakak suffers for lack of validation through fielded systems or experimentation. Although the results are intuitively appealing they are unsubstantiated in application.

A third problem is of particular import to analytical modeling. That is, many system attributes which may be important to distributed system survivability are difficult to measure. Foremost among these

features are those which pertain to software. In earlier fault tolerant systems, software was a minor consideration. Currently software is a primary consideration, and the necessity to incorporate software factors into system evaluation is unavoidable (12).

Whereas some sciences in their early stages are inexact, other sciences are inherently inexact (13). For a philosophical discussion of exact versus inexact sciences, see reference (13). Software is not subject to static standards or metrics but rather must be indirectly described in terms of those attributes which can be measured or observed. Among these attributes are requirements measured in terms of instructions to be executed, storage demands, input/output data rates and application module quantity, size and connectivity. These measurements are used to form the basis for prediction of software related phenomena like development, maintainability and life cycle cost. In that these measurements are rarely derived from first principles, it is unlikely that we can undertake their explanation and prediction in a wholly quantitative manner without imposing severe constraints on the level of complexity to be addressed (7).

Empiricism offers the opportunity to develop statistical laws which can serve several purposes (13). First, it can enhance our understanding of phenomena and provide a basis for prediction or decision making. Second, it can point to areas in which purely analytical investigation can be productive and, third, it can provide a mechanism for validation of analytical models. In addition empiricism does not inherently prejudge research findings and thus establishes essential objectivity.

For these reasons, this dissertation proposes an alternative approach to survivability explanation and estimation. First, a simulation approach to survivability experimentation is taken. Second, a broad spectrum of distributed system attributes are examined. The attributes fall into three general categories namely; those that describe the distributed system capabilities and topology, those that describe the application system topology and requirements and those that describe the distribution and redistribution policies which map the software onto the hardware. In addition, software is permitted to degrade gracefully, that is reduce its resource demands, in order to accommodate degradation of the distributed network. The objective of this dissertation is to provide a foundation for the quantification of operational survivability in gracefully degrading distributed processing systems which can be empirically tested and generally applied.

CHAPTER III

APPROACH

As indicated in the preceding chapter, this research proposes that survivability is a function of a number of attributes. These attributes fall into three general categories, i.e., those that describe the distributed network, those that describe the application system, and those that describe the distribution policy. Further, this dissertation proposes that none of these categories taken alone serve adequately to explain or predict the survivability of a given system. The method of this research is to investigate the relationship between a number of attributes of distributed systems and survival of those systems in the face of increasing degradation of network resources.

Several alternative approaches to this investigation have been considered. The approach must facilitate manipulation of a number of factors pertaining to distributed networks, application systems and distribution approaches for the purpose of analysis. Perhaps the most likely alternative from the point of flexibility is an analytical model. However, the constituents of distributed systems such as routing, resource allocation and task assignment when individually subjected to analytical study present difficult and complex problems. Among these problems are measurement and computability problems. Many system attributes are difficult to describe quantitatively such as reconfiguration options or decisions, resource capabilities and

execution constraints. When quantitative description is possible, it often shows exponential growth with increases in system size. It follows, therefore, that a system comprising many of these constituents would be correspondingly more difficult to represent analytically (19,14). Further, given an analytical approach a decision must be made to either oversimplify the distributed processing problem or address the potential problem of intractability.

The second major alternative is empirical. If an experimental approach is to be used, a choice must be made first between field versus laboratory experimentation. Since instances of operational distributed processing systems are few, opportunities for field experiments at this time are commensurately limited. For these reasons laboratory experimentation was selected as the most viable approach for this research. Laboratory experimentation via simulation permits the representation and control of factors in the field environment which are controllable and many which are not. The degree to which simulation can present a "true" picture of that which it simulates is dependent upon the level of understanding which exists of the components, actions and interactions of the simulated phenomena. Simulation differs from analytical modeling in that the relationships between the proposed factors need not be stated in an explicitly quantitative fashion. Once a simulation facility exists, the subsequent capability for controlled replication takes major advantage over field experimentation. Also, the ability to monitor and track operational conditions, decisions, and intermediate actions can be considerably easier than in field situations.

Computer simulation has been chosen for examination of the problem of distributed system survivability because of the large number of variables which must be used, the complexity of their manipulation and the magnitude of the possible instantiations of the variable values.

Computer simulation, allows a "systems view" of distributed systems to be presented. The first objective of this research is to determine those factors which have an important effect on distributed system survivability and can be used in the development of a measure or model of survivability. Second, this research proposes to provide a simulation approach to distributed system comparison. Since experimentation via simulation is similar in many ways to field test, traditional experimental methods and analysis techniques should apply. One intent of this research is, in fact, to demonstrate the applicability of standard experimental design and analysis techniques to topics in computer science.

The following provides an introduction and discussion of the experimental approach to be used. Specifics of implementation in the present investigation are given in Chapter IV.

The literature repeatedly paints the picture of distributed systems as a very complex one comprised of numerous orthogonal and interrelated factors such as levels of hardware, control and data base decentralization (8,15). One goal of this work is to discover the subset of active factors which is important to survivability. The process of factor selection is called factor screening. Factor screening must take place with a comprehensive set of active factors,

because omission of an important active factor can introduce consequences such as bias in the analysis and conclusions drawn from the experiment. Inclusion of negligible factors may, on the other hand, be unnecessarily resource consumptive and introduce sufficient *noise in the data such that important* effects are difficult to *recognize*.

Factor screening methods can be introduced either during the design and development of the simulation model or upon completion of the simulator. When employed at early stages, as is proposed here, factor screening will affect the choice of variables and variable levels *in the model. The overall impact will be to simplify the* structure *of the final model and sharpen* discription of specific effects (16).

Let us suppose that there are a number of controllable factors in the simulation experiment, call these $X_1, X_2, \ldots, X_K$ and two response variables S and P. Since S, survivability, for now is assumed to be two-valued and be a function of some range in P, performance, such that

$$S = \begin{cases} 1 & \text{if } P \leq n \\ 0 & \text{otherwise} \end{cases} \qquad (3\text{-}1)$$

we can proceed as though there is but one response, P. Further let us assume *that the simulator is structured such that the response can be* expressed in the form

$$P = f\,(X_1, X_2, \ldots, X_K) + \epsilon$$

where f is a function that determines the mean value of P, and ϵ represents error such that, $E(\epsilon)=0$, the expected value of ϵ is zero. Initially, it is assumed that f is linear in the unknown parameters, coefficients, *that relate the response*, P, to the *factors*, $X_1, X_2, \ldots, X_K$. One possible model is

$$P = \beta_0 + \sum_{i=1}^{K} \beta_i X_i + \epsilon \qquad (3\text{-}2)$$

where $\beta_0, \beta_1, \ldots, \beta_K$ are unknown parameters. Here β_0 is the intercept and $\beta_1, \beta_2, \ldots, \beta_K$, the coefficients.

To use this system to conduct an experiment, the levels of each factor must be chosen and the simulation run on the full set or some subset of the *factor level combinations*. The selection of the number of factor levels to be used and their spacing is extremely important. Since, in this research, as in many factor screening experiments, we are trying to determine the relative effect of a factor and not develop a highly precise predictive or interpolative equation, the number of factor levels or values to be tested will be small, two or four. The "effect" of a factor is described as the change in the response observed *as a result of a change in levels of the factor*. This direct *cause-effect relationship* between a single factor and the response is called a "main" effect.

Factor screening experiments fall into two major categories, full factorial experiments and fractional factorial experiments. The most efficient full factorial design is the 2^K factorial design which

comprises K factors each having two levels. The statistical model generated for a 2^K full factorial design would include K main effects, $\binom{K}{2}$ two-factor interactions, $\binom{K}{3}$ three-factor interactions, $\binom{K}{4}$four-factor interactions, etc. and one K-factor interaction. In total the 2^K design would describe 2^K-1 effects.

The term treatment combination is used to refer to the aggregate of factor settings of all factors as designated for a given experiment run or case. One system of notation frequently used to denote individual factor levels, uses + and - signs to designate high and low or alternate levels of the factor. Thus, a treatment combination for a four factor experiment on factors X_1, X_2, X_3, X_4 might be - + + - indicating that factors X_1 and X_4 are at their low setting and factors X_2 and X_3 at their high setting.

The total number of experiment runs required in a 2^K full factorial design given small values of K such as 5 or 6 is 32 and 64 respectively. The magnitude of this number grows exponentially with K. Since resources are usually limited, the number of replicates that the experimenter can employ may be restricted. Frequently, available resources will only allow a single replicate of the design to be run, unless the experimenter is willing to omit some of the original factors.

With only a single replicate of the 2^K it is impossible to compute an estimate of experimental error, that is, a mean square for error. Thus, hypotheses concerning main effects and interactions cannot be tested. However, the usual approach to the analysis of a single replicate of a 2^K full factorial design is to assume that

certain higher-order interactions are negligible (21). The statistical analysis of these designs by either Yates' tabular algorithm or a regression approach may be used to estimate the effects. Since this is a factor screening experiment, our interest will be confined to detecting main effects and 2-factor interactions. We can, therefore, either use the higher-order effects as an estimate of error, or as the basis of developing a more efficient design via fractional replication. By assuming that certain high-order interactions are negligible, information on main effects and low-order interactions may be obtained by running only a fraction of the complete factorial experiment. These fractional factorial designs are widely used in research and have major applications in factor screening (21).

In a 2^{K-P} fractional factorial design, only a fraction, 1/2 P, of the 2^K treatment combinations are actually run. A fraction of the 2^K design containing 2^{K-P} runs is called a 1/2 P fraction of the 2^K full factorial design, or a 2^{K-P} fractional factorial design. The design proposed in this research is a regular fraction, that is, estimates of the effects are orthogonal. The effects may be estimated by generating the contrast for any factor using the table of + and - signs for that design which is equivalent to the regression approach outlined above. There are several commonly used methods of constructing these designs.

The particular 2^{K-P} fractional factorial design to be used in this research is of resolution V usually expressed as 2^{K-P}_V. In a resolution V design an unconfounded estimation of all main effects and two factor interactions is obtained. Three factor interactions and higher will be confounded or aliased in such a way that isolation of

particular effects is not possible. The higher the resolution of the fractional factorial design, the greater the information obtained concerning higher order interactions. The higher the resolution, the closer the fractional factorial design comes to a full factorial design and consequently the greater the number of experiment runs required. It follows that as the size of K increases, the number of experiment runs required to meet higher resolution designs is directly effected. Selection of the appropriate design resolution is an important part of initial research considerations. For further information on 2^{K-P}_{V} fractional factorial designs the reader is referred to two papers by Box and Hunter (6,9).

CHAPTER IV

PROCEDURE

The rationale for the simulation approach to model design was presented in Chapter III. The objective of this simulation is to facilitate determination of those factors which have an important effect on distributed system survivability and can be used to develop a measure or index of survivability. In addition it is anticipated that this simulation approach will be used to compare distributed processing systems. A discussion of the initial simplifying assumptions, variable selection and quantification is provided below. In addition, the basic 2^{K-P} fractional factorial resolution V experimental design is described, the experimental approach outlined and the basic structure of the simulator is presented.

Assumptions

To effect a simulation which comprises adequate variables to represent a realistic distributed processing system and sufficiently well specified to permit experimentation, three simplifying assumptions on the distributed system attributes are used. As experimentation with the proposed simulator proceeds it may be possible to relax some of these constraints. The initial assumptions are discussed below.

1. All software support resources and application software is accessible by all processing nodes. It is, of course, not likely that these resources are all equally easy to access;

however, the complexity of accessibility will not be addressed in this research. This assumption is made so that the issue of application and support software transfer from one node to another need not be addressed. This assumption is realistic for application and support software on homogeneous networks but falls short when changing data bases are considered. This assumption as it relates to data bases will be relaxed in future experiments.

2. Loss of communication links alone is not considered in these experiments. Loss of a node will, of course, eliminate all links connecting to that node. The effects of link loss is a very complex problem which continues to be extensively researched in connection with various types of networks (9,10,11,1). The loss of individual links can be readily incorporated into the experiment setting proposed for this research. In essence since the removal of a node implies the removal of all adjoining links, creation of an artificial node representing a given link and the subsequent removal of that node will have the same effect as removal of the original link.

3. The simulator has control over vulnerability. The vulnerability and criticality of individual processing nodes are very important considerations for many applications and can be incorporated in the proposed simulator at a later

date. Presently, however, omitting these factors allows us to focus on the structural features of distributed systems which effect operational survivability. Both static and dynamic vulnerability and criticality attributes will be added in later experiments.

<u>Experiment Factors and Factor Levels</u>

A distributed processing system is a computer network composed of two or more autonomous processing and memory components working together to serve a common application. A *gracefully degrading system* is a multiple processor system which provides a high quality of service by reconfiguring the system or network or by reallocating resources when a fault is detected. Operational survivability, then, is an attribute describing the degree to which a distributed processing system can gracefully degrade. The objectives of this research are to make our understanding of survivability a quantitative one and to develop a model or set of models with which we can evaluate and predict operational survivability and performance. The survivability index can be expressed as a simple function of level of performance. In this research, performance can have one of four values depending on the level to which application system requirements are satisfied.

Performance value

1	indicates normal or satisfactory application system performance
2,3	indicate satisfactory degraded application performance
4	indicates unsatisfactory application system

performance.

Satisfactory degraded application system performance refers to the success of the distributed system to adjust to a loss of distributed network resources by a reduction in application system requirements. The survivability index will have either the value "1" indicating that a given distributed system is survivable or 0 indicating that the system is not survivable according to the following

$$\text{Survivability Index} = \begin{cases} 1 \text{ if Performance} \leq 3 \\ 0 \text{ otherwise} \end{cases} \tag{4-1}$$

The value assigned to performance can be expressed as a function of a number of attributes

$$P = f(Z_1, Z_2, \ldots, Z_K) \tag{4-2}$$

such that attributes Z_1, Z_2, ..., Z_K describe features of the distributed network, application system and distribution policy. The Zs represent features of the distributed system which are manipulated or controlled.

The parameters that will be controlled in the proposed 2_V^{K-P} Fractional Factorial design are presented in Table 1.

Table 1. Experimental Factors and Factor Levels

	Factor	Levels
Z1	Type of Distributed Processing Topology	a. STAR b. RING c. NETWORK d. ARRAY
Z2	Number of Nodes	a. 4 b. 10
Z3	Node Processing Speed	a. 500 KOPS b. 10 MOPS
Z4	Node Memory Capacity	a. 128 KBYTES b. 2 MBYTES
Z5	Connectivity of Application System	a. Low b. High
Z6	Number of Application Modules	a. 4 b. 16
Z7	Average Module Processing Requirements	a. 10% Node Processing Capacity b. 50% Node Processing Capacity
Z8	Average Module Memory Requirements	a. .1 Node Memory Capacity b. .8 Node Memory Capacity
Z9	Average Frequency of Module to Module Interaction (Δ of thousand message set ups)	a. Low b. High
Z10	Distribution/Redistribution Strategy	a. Random b. Uniform c. Packed d. Optimal Spare
Z11	Percent of Nodes Eliminated	a. 10% b. 30% c. 50% d. 80%

Note: For further description of these factors see Appendix A.

Table 2. below shows the correspondence between the eleven variables in the preceding chart and the pseudo factors used in the 2^{K-P} design proposed here. The pseudo factors are used to create 2 two-level factors to represent each four level factor.

Table 2. Experiment Factors and Pseudo-factors

ORIGINAL FACTORS	NO LEVELS	PSEUDO-FACTORS	LABELS
Z_1	4	X_1)	A
		X_2) *	B
Z_2	2	X_3	C
Z_3	2	X_4	D
Z_4	2	X_5	E
Z_5	2	X_6	F
Z_6	2	X_7	G
Z_7	2	X_8	H
Z_8	2	X_9	J
Z_9	2	X_{10}	K
Z_{10}	4	X_{11})	L
		X_{12}) *	M
Z_{11}	4	X_{13})	N
		X_{14}) *	O

* Considered Together

Thus, according to Table 2. it is apparent that factors Z_1, Z_{10}, and Z_{11} are decomposed to 2 two-level pseudo factors. When designing experiment runs, pairs of pseudo factors are considered together.

Let us consider now the design of a fourteen factor experiment. Since this research is concerned with both main effects and two factor

interactions, the Resolution V design is considered. This design provides the desired clarity of main effects and two factor interactions. Implementation of this design for our fourteen factor experiment proceeds as follows. There are fourteen main effects and $\binom{14}{2}$ or 91 possible two factor interactions which gives a total of 105 effects. Taking the next higher power of two, 2^7 indicates that 128 experimental runs would have to be made to cover all the effects of interest. Thus, rather than 2^{14} runs only 2^{14-7} runs, or 1/128 of the total possible combinations need be tried.

Next, it is necessary to describe the individual runs or treatment combinations which must be executed. To construct the chart of experiment runs shown in Table 3. first the plus and minus levels for a full 2^7 design in A, B, C, D, E, F, and G is established. Letters here represent factors. The levels for the 7 remaining factors are generated using interactions of the original seven factors as follows:

H=ABCG, J=BCDE, K=ABDF, L=AEFG,

M=CDFG, N=ACEFG, and O=BDEFG.

Thus, the generating relations for this design are

I=ABCGH, I=ABCDEJ, I=ABDFK, I=AEFGL,

I=CDFGM, I=ACEFGN, and BDEFGO.

Table 3.

2(K-P) FRACTIONAL FACTORIAL DESIGN

SURVIVABILITY SIMULATOR

EXPERIMENT RUN DESCRIPTIONS

ORIGINAL Z-FACTORS

	A	B	C	D	E	F	G	H=ABCG	J=BCDE	K=ACDF	L=ABFG	M=CDFG	N=ACEFG	O=BDEFG
RUN	1	2	3	4	5	6		7	8	9	10		11	
[illegible]	-	-	-	-	-	-	-	+	+	+	+	+	-	-
[illegible]	+	-	-	-	-	-	-	-	+	-	-	+	+	-
[illegible]	-	+	-	-	-	-	-	-	-	-	+	+	-	+
[illegible]	+	+	-	-	-	-	-	+	-	+	-	+	+	+
[illegible]	-	-	+	-	-	-	-	-	-	+	+	-	+	-
[illegible]	+	-	+	-	-	-	-	+	-	-	-	-	-	-
[illegible]	-	+	+	-	-	-	-	+	+	-	+	-	+	+
[illegible]	+	+	+	-	-	-	-	-	+	+	-	-	-	+
[illegible]	-	-	-	+	-	-	-	+	-	-	+	-	-	+
[illegible]	+	-	-	+	-	-	-	-	-	+	-	-	+	+
[illegible]	-	+	-	+	-	-	-	-	+	+	+	-	-	-
[illegible]	+	+	-	+	-	-	-	+	+	-	-	-	+	-
[illegible]	-	-	+	+	-	-	-	-	+	-	+	+	+	+
[illegible]	+	-	+	+	-	-	-	+	+	+	-	+	-	+
[illegible]	-	+	+	+	-	-	-	+	-	+	+	+	+	-
[illegible]	+	+	+	+	-	-	-	-	-	-	-	+	-	-
[illegible]	-	-	-	-	+	-	-	+	-	+	-	+	+	+
[illegible]	+	-	-	-	+	-	-	-	-	-	+	+	-	+
[illegible]	-	+	-	-	+	-	-	-	+	-	-	+	+	-
[illegible]	+	+	-	-	+	-	-	+	+	+	+	+	-	-
[illegible]	-	-	+	-	+	-	-	-	+	+	-	-	-	+
[illegible]	+	-	+	-	+	-	-	+	+	-	+	-	+	+
[illegible]	-	+	+	-	+	-	-	+	-	-	-	-	-	-
[illegible]	+	+	+	-	+	-	-	-	-	+	+	-	+	-
[illegible]	-	-	-	+	+	-	-	+	+	-	-	-	+	-
[illegible]	+	-	-	+	+	-	-	-	+	+	+	-	-	-
[illegible]	-	+	-	+	+	-	-	-	-	+	-	-	+	+
[illegible]	+	+	-	+	+	-	-	+	-	-	+	-	-	+
[illegible]	-	-	+	+	+	-	-	-	-	-	-	+	-	-
[illegible]	+	-	+	+	+	-	-	+	-	+	+	+	+	-
[illegible]	-	+	+	+	+	-	-	+	+	+	-	+	-	+
[illegible]	+	+	+	+	+	-	-	-	+	-	+	+	+	+
[illegible]	-	-	-	-	-	+	-	+	+	-	-	-	+	+
[illegible]	+	-	-	-	-	+	-	-	+	+	+	-	-	+
[illegible]	-	+	-	-	-	+	-	-	-	+	-	-	+	-
[illegible]	+	+	-	-	-	+	-	+	-	-	+	-	-	-
[illegible]	-	-	+	-	-	+	-	-	-	-	-	+	-	+
[illegible]	+	-	+	-	-	+	-	+	-	+	+	+	+	+
[illegible]	-	+	+	-	-	+	-	+	+	+	-	+	-	-
[illegible]	+	+	+	-	-	+	-	-	+	-	+	+	+	-
[illegible]	-	-	-	+	-	+	-	+	-	+	-	+	+	-

Table 3 continued.

2(K-P) FRACTIONAL FACTORIAL DESIGN

SURVIVABILITY SIMULATOR

EXPERIMENT RUN DESCRIPTIONS

ORIGINAL Z-FACTORS

	A	B	C	D	E	F	G	H=ABCG	J=BCDE	K=ACDF	L=ABFG	M=CDFG	N=ACEFG	O=BDEFG
RUN	1	2	3	4	5	6		7	8	9	10		11	
42	+	-	-	+	-	+	-	-	-	-	+	+	-	-
43	-	+	-	+	-	+	-	-	+	-	-	+	+	+
44	+	+	-	+	-	+	-	+	+	+	+	+	-	+
45	-	-	+	+	-	+	-	-	+	+	-	-	-	-
46	+	-	+	+	-	+	-	+	+	-	+	-	+	-
47	-	+	+	+	-	+	-	+	-	-	-	-	-	+
48	+	+	+	+	-	+	-	-	-	+	+	-	+	+
49	-	-	-	-	+	+	-	+	-	-	+	-	-	-
50	+	-	-	-	+	+	-	-	-	+	-	-	+	-
51	-	+	-	-	+	+	-	-	+	+	+	-	-	+
52	+	+	-	-	+	+	-	+	+	-	-	-	+	+
53	-	-	+	-	+	+	-	-	+	-	+	+	+	-
54	+	-	+	-	+	+	-	+	+	+	-	+	-	-
55	-	+	+	-	+	+	-	+	-	+	+	+	+	+
56	+	+	+	-	+	+	-	-	-	-	-	+	-	+
57	-	-	-	+	+	+	-	+	+	+	+	+	-	+
58	+	-	-	+	+	+	-	-	+	-	-	+	+	+
59	-	+	-	+	+	+	-	-	-	-	+	+	-	-
60	+	+	-	+	+	+	-	+	-	+	-	+	+	-
61	-	-	+	+	+	+	-	-	-	+	+	-	+	+
62	+	-	+	+	+	+	-	+	-	-	-	-	-	+
63	-	+	+	+	+	+	-	+	+	-	+	-	+	-
64	+	+	+	+	+	+	-	-	+	+	-	-	-	-
65	-	-	-	-	-	-	+	-	+	+	-	-	+	+
66	+	-	-	-	-	-	+	+	+	-	+	-	-	+
67	-	+	-	-	-	-	+	+	-	-	-	-	+	-
68	+	+	-	-	-	-	+	-	-	+	+	-	-	-
69	-	-	+	-	-	-	+	+	-	+	-	+	-	+
70	+	-	+	-	-	-	+	-	-	-	+	+	+	+
71	-	+	+	-	-	-	+	-	+	-	-	+	-	-
72	+	+	+	-	-	-	+	+	+	+	+	+	+	-
73	-	-	-	+	-	-	+	-	-	-	-	+	+	-
74	+	-	-	+	-	-	+	+	-	+	+	+	-	-
75	-	+	-	+	-	-	+	+	+	+	-	+	+	+
76	+	+	-	+	-	-	+	-	+	-	+	+	-	+
77	-	-	+	+	-	-	+	+	+	-	-	-	-	-
78	+	-	+	+	-	-	+	-	+	+	+	-	+	-
79	-	+	+	+	-	-	+	-	-	+	-	-	-	+
80	+	+	+	+	-	-	+	+	-	-	+	-	+	+
81	-	-	-	-	+	-	+	-	-	+	+	-	-	-
82	+	-	-	-	+	-	+	+	-	-	-	-	+	-
83	-	+	-	-	+	-	+	+	+	-	+	-	-	+
84	+	+	-	-	+	-	+	-	+	+	-	-	+	+
85	-	-	+	-	+	-	+	+	+	+	+	+	+	-

Table 3 continued.

2(k-p) FRACTIONAL FACTORIAL DESIGN

SURVIVABILITY SIMULATOR

EXPERIMENT RUN DESCRIPTIONS

ORIGINAL Z-FACTORS

RUN	A	B	C	D	E	F	G	H=ABCG	J=BCDF	K=ACDE	L=ABEF	M=CDEG	N=ACEFG	O=BDEFG
	1		2	3	4	5	6	7	8	9	10		11	
86	+	-	+	-	+	-	+	-	+	-	-	+	-	-
87	-	+	+	-	+	-	+	-	-	-	+	+	+	+
88	+	+	+	-	+	-	+	+	-	+	-	+	-	+
89	-	-	-	+	+	-	+	-	+	-	+	+	-	+
90	+	-	-	+	+	-	+	+	+	+	-	+	+	+
91	-	+	-	+	+	-	+	+	-	+	+	+	-	-
92	+	+	-	+	+	-	+	-	-	-	-	+	+	-
93	-	-	+	+	+	-	+	+	-	-	+	-	+	+
94	+	-	+	+	+	-	+	-	-	+	-	-	-	+
95	-	+	+	+	+	-	+	-	+	+	+	-	+	-
96	+	+	+	+	+	-	+	+	+	-	-	-	-	-
97	-	-	-	-	-	+	+	-	+	-	+	+	-	-
98	+	-	-	-	-	+	+	+	+	+	-	+	+	-
99	-	+	-	-	-	+	+	+	-	+	+	+	-	+
100	+	+	-	-	-	+	+	-	-	-	-	+	+	+
101	-	-	+	-	-	+	+	+	-	-	+	-	+	-
102	+	-	+	-	-	+	+	-	-	+	-	-	-	-
103	-	+	+	-	-	+	+	-	+	+	+	-	+	+
104	+	+	+	-	-	+	+	+	+	-	-	-	-	+
105	-	-	-	+	-	+	+	-	-	+	+	-	-	+
106	+	-	-	+	-	+	+	+	-	-	-	-	+	+
107	-	+	-	+	-	+	+	+	+	-	+	-	-	-
108	+	+	-	+	-	+	+	-	+	+	-	-	+	-
109	-	-	+	+	-	+	+	+	+	+	+	+	+	+
110	+	-	+	+	-	+	+	-	+	-	-	+	-	+
111	-	+	+	+	-	+	+	-	-	-	+	+	+	-
112	+	+	+	+	-	+	+	+	-	+	-	+	-	-
113	-	-	-	-	+	+	+	-	-	-	-	+	+	+
114	+	-	-	-	+	+	+	+	-	+	+	+	-	+
115	-	+	-	-	+	+	+	+	+	+	-	+	+	-
116	+	+	-	-	+	+	+	-	+	-	+	+	-	-
117	-	-	+	-	+	+	+	+	+	-	-	-	-	+
118	+	-	+	-	+	+	+	-	+	+	+	-	+	+
119	-	+	+	-	+	+	+	-	-	+	-	-	-	-
120	+	+	+	-	+	+	+	+	-	-	+	-	+	-
121	-	-	-	+	+	+	+	-	+	+	-	-	+	-
122	+	-	-	+	+	+	+	+	+	-	+	-	-	-
123	-	+	-	+	+	+	+	+	-	-	-	-	+	+
124	+	+	-	+	+	+	+	-	-	+	+	-	-	+
125	-	-	+	+	+	+	+	+	-	+	-	+	-	-
126	+	-	+	+	+	+	+	-	-	-	+	+	+	-
127	-	+	+	+	+	+	+	-	+	-	-	+	-	+
128	+	+	+	+	+	+	+	+	+	+	+	+	+	+

To determine *the level for each 2 level factor simply interpret* the corresponding plus or minus sign. To determine the level for four level factors the following set of index tables will be used.

a) ORIGINAL FACTOR Z_1 LEVELS
PSEUDO FACTORS A,B

		A	
		LOW	HIGH
B	LOW	- - Z_{1-c}	- + Z_{1-b}
	HIGH	+ - Z_{1-c}	+ + Z_{1-d}

b) ORIGINAL FACTOR Z_{10} LEVELS
PSEUDO-FACTORS L,M

		L	
		LOW	HIGH
M	LOW	- - Z_{10-a}	- + Z_{10-b}
	HIGH	+ - Z_{10-c}	+ + Z_{10-d}

c) ORIGINAL FACTOR Z_{11} LEVELS
PSEUDO-FACTORS N,O

		N	
		LOW	HIGH
O	LOW	- - Z_{11-a}	- + Z_{11-b}
	HIGH	+ - Z_{11-c}	+ + Z_{11-d}

Figure 1. Index Tables for Four Level Factors

Using Tables 1,2 and 3 and Figure 1, run #1 of this experiment would be composed as follows:

Table 4. Interpretation of Example Treatment Combination

RUN #1

Z_1	STAR
Z_2	4 NODES
Z_3	500 KOPS PROCESSING
Z_4	128 KBYTES MEMORY CAPACITY
Z_5	LOW APPLICATION SYSTEM CONNECTIVITY
Z_6	LOW # APPLICATION MODULES
Z_7	50,000 KOPS/EXECUTION AVERAGE MODULE PROCESSING REQUIREMENTS
Z_8	AVERAGE MODULE USES .8 OF NODE MEMORY CAPACITY
Z_9	HIGH # OF MESSAGE SET UPS
Z_{10}	OPTIMAL SPARE DISTRIBUTION
Z_{11}	10% NODES ELIMINATED

SURSIM Survivability Simulator

SURSIM is a simulator which facilitates the investigation of the concept of survivability in gracefully degrading systems. It examines distributed system resources, processing nodes and associated links, which can be lost before a given application system required to execute on that distributed system must function in a degraded mode or experience failure.

The Survivability Simulator depicted in Figure 2 shows the *function and flow* of the system. SURSIM accepts the description of arbitrary application system topologies and requirements, and distributed system *topologies and capabilities, and using predetermined* configuration and reconfiguration strategies exercises the hardware/ software systems through a sequence of *hits* or *node losses which reduce* the capability of the distributed processing system. Effects of configuration modification and capability reduction on application system performance is analyzed. Based on this analysis the application system is reconfigured or the distributed system is further mutated. The simulator continues to iterate reconfigurations and mutations while logging performance and configuration data until the distributed system fails, i.e. the application system can no longer function on the distributed system at an acceptable level.

Within the simulator, the application system and distributed processing network are represented as graphs. For the application system the vertices represent program modules and the edges represent module interaction. For the distributed processing network the vertices represent processing nodes and the edges represent

communication links. Application system requirements are described in terms of *module memory requirements*, *processing requirements*, *frequency* of execution, frequency of module to module interaction, and module criticality. The capability to systematically reduce application system demands according to some apriori defined policy exists. The degree to which procedures of the application system degradation policy are implemented depends upon the degradation level of the distributed network. Distributed system capabilities are described in terms of node processing speed, memory size, and communications capacity. Several different approaches to task assignment are simulated. *Via* these policies the application system is *mapped onto the distributed* processing network. *This is a graph mapping* which is performed *according to one of four* policies. The four policies are 1.) random distribution, 2.) uniform distribution, 3.) packed distribution and 4.) the optimal-spare distribution. In the random distribution, application system modules are randomly assigned to processors. This will be repeated until all modules have been assigned to nodes or the policy fails to construct a map. If the application module and communication burden exceed that of the node selected, assignment will not be made. In the uniform distribution, application system modules are assigned to nodes such that each node has as near the same operating demands as possible. In the packed distribution, application system modules are assigned to a designated processor until it reaches maximum capacity after which modules are assigned to *the "next"* processor, etc. *In the optimal-spare distribution*, application system modules *are assigned to the* distributed processing system explicitly by

the system designer. Each node being assigned application tasks has a spare queue indicating the sequence of backup or spare nodes which will be actived should the former fail. This distribution approach takes into account the requirement of certain application modules for processing nodes with special I/O devices such as sensors and actuators.

The performance *analyzer performs a comparison of application* system *requirements to the specific* distributed system capabilities assigned to it. For each node in the distributed system a comparison is made between the node capability and the application system requirements of all modules assigned to it. For example, if the memory capacity of a node less the memory requirements of all modules assigned to it gives a negative result, performance is considered unsatis-factory. Likewise, if the processor demands exceed the processor capability performance is considered unacceptable. The ability of communications links to meet expected demands is similarly determined by accessing resource saturation. Should the performance analyzer indicate that performance in the current application system/distributed system configuration is *satisfactory in all categories*, *the distributed* system topology *will be further mutated*, *otherwise the* application system *reconfiguration segment* of the simulator will be instantiated.

The function of the distributed system topology mutator is to systematically eliminate nodes and their associated links until the distributed system topology is such that "satisfactory" application system performance cannot be achieved. The approach is as follows. First, each individual node and its associated links is removed, then

all possible combinations of two nodes, then three node combinations, etc., until all possible mutations of the distributed system topology have been exercised. The loss of multiple nodes is treated as though these losses occur simultaneously; however, a more advanced form of the simulator should be able to take into account history dependence of failures.

The function of the application system reconfiguration segment of the simulator is to carry out the distribution policy in effect and institute the degradation procedures as necessary. This simulator segment is called into operation when the application system performance analyzer indicates an unsatisfactory level of application system performance. An attempt will be made to reconfigure the application system using whatever distribution policy is in effect to bring the system to an acceptable performance level. Should the reassignment efforts fail to bring performance to the desired level the apriori stated procedures for software degradation will be imposed. Following instantiation of each degradation procedure, performance will be reevaluated. This process is iterated until satisfactory degraded performance is achieved or all degradation procedures have been implemented. In the latter case, the distributed system will have failed to meet the application system performance requirements in normal or degraded mode and consequently, will be considered inoperable.

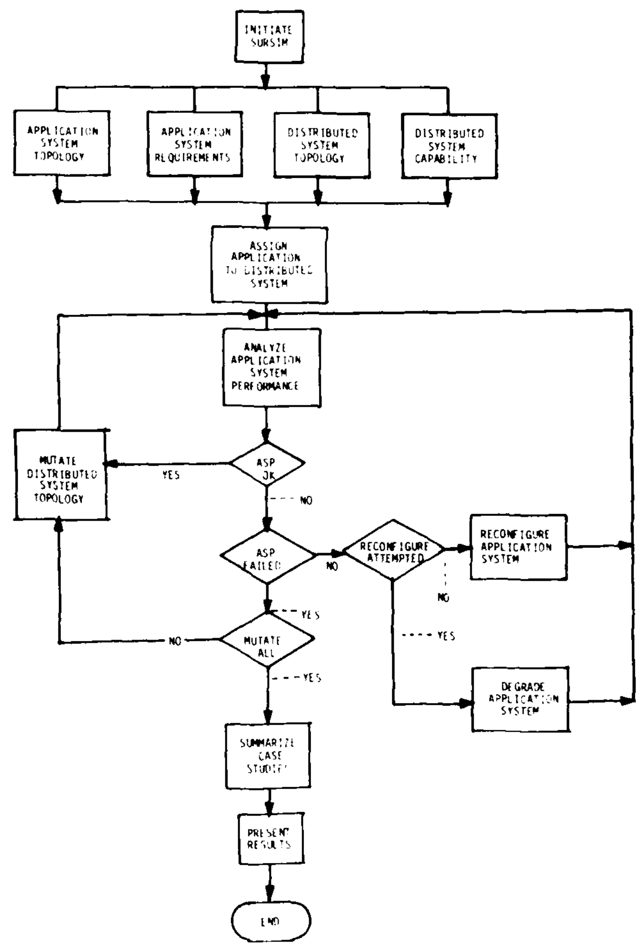
The performance analysis routine in part determines the class of service being provided. Two categories of acceptable service may exist: normal or degraded, and one category of unacceptable service:

failed. There are two ways in which the application system can go from normal to degraded mode. One is to "degrade" or reduce performance requirements. This essentially means that the application modules will continue to perform all their current functions but at a slower rate. The other means of degradation is to "cripple" the application system or purge designated application modules. To what level processing or interaction requirements are reduced or which modules are purged and in what order is determined apriori by the application system designer. This information is input to the simulator.The distributed system continues to be systematically changed by the distributed system topology mutator until application system performance has degenerated to an unacceptable level or all mutations of the distributed network have been exercised.

Information collected by the simulator falls into three categories; 1.) status of controlled factors, 2.) status of indirectly controlled factors, and 3.) derived measures. Controlled factors include; type of distributed system topology, size of network or number of nodes; node processing speed, memory and communications capacity; application system size, connectivity and interaction, processing, and memory requirements; distribution strategy and extent of distributed system degradation (mutation). Indirectly controlled factors include connectivity of the distributed system topology, global resource capacity (processing, memory, communications) and available resource capabilities (processing, memory, and communications). Derived information includes a variety of resources:requirements ratios and consistency measures.

SURSIM, the survivability simulator, has been implemented in FLECS, FORTRAN Language with Extended Control Structures, on a Digital Equipment VAX 11/780 and is now being used as a tool for experimentation on a variety of distributed systems.

Figure 2. Survivability Simulator Flow Diagram



CHAPTER V

SIMULATOR RESULTS

Output generated by the simulator for the 128 designed experiment runs and 300,000 subcases fall into two categories. The first type of output is strictly descriptive of the cases run. The second type of output is a log of operational data collected for each of the cases and subcases. Discussion and examples of the output follow.

Descriptive Output

Table 5 is a chart generated by the simulator for each of the 128 designed experiment runs. It presents a string of + and - signs which designate the factor level of each of the original 11 factors for that case followed by an English language interpretation of the factor levels. Table 6 is a representation of the application system interactions for this case. Table 7 lists the respective application system requirements for each application module. Table 8 presents the degradation procedures to be followed in the event that the application system cannot function in a satisfactory manner on the distributed system network given its current configuration and application system assignments. Table 9 presents an interaction incidence matrix which describes the topology of the distributed network for this case. The queue of start nodes lists all the potentially unique start nodes for the topology being studied. Table 10 represents the capability of each node in the distributed system and its connection to other nodes. Tables 11 and 12 describe the remaining resources after initial

assignment of the application system to the distributed system and module to node assignments respectively.

Operational Data Logged

Output of the simulator which is descriptive of control factors or records operational data is logged for later analysis. Examples of this output are shown in Table 13. Definition of logged data items follows:

R_1	S	- Response variable - survivability 1 = survival 2 = failure
R_2	P	- Response variable - performance 1 = normal satisfactory performance 2,3 = degraded satisfactory performance 4 = unsatisfactory or failed performance
Z_1	DIS	- Distributed system topology
Z_2	N	- Number of nodes in the distributed network
Z_3	NPS	- Node processing speed
Z_4	NMC	- Node memory capacity
Z_5	C	- Node communications capacity
Z_6	M	- Number of application system modules
Z_7	MPR	- Average application module processing requirements
Z_8	NMR	- Average application module memory requirements
Z_9	MIF	- Average module to module interaction frequency
Z_{10}	POL	- Distribution policy
Z_{11}	PCT	- Percent nodes lost
Z_{12}	SN	- Start node

Z_{13}	IAS	- Initial assignment result (success or failure)
Z_{14}	R	- Reconfiguration status (not attempted, success, failure)
Z_{15}	D	- Degradation status (not attempted, step if invoked)
Z_{16}	A%	- NA
Z_{17}	=	- NA
Z_{18}	NLN	- Number of lost nodes
Z_{19}-Z_{28}	1-10	- Labels of actual nodes lost
	RUN	- Identifier for a given execution of SURSIM designating the designed experiment run in effect.
	CASE	- Identifier for a given subcase of SURSIM designating the unique mutation in effect.
Z_{29}	GLP	- Global processing capacity
Z_{30}	GLM	- Global memory capacity
Z_{31}	GLC	- Global communications capacity
Z_{32}	AVP	- Available processing capacity after initial assignment
Z_{33}	AVM	- Available memory capacity after initial assignment
Z_{34}	AVC	- Available communications capacity after initial assignment
Z_{35}	DSAVP	- Available processing at end of subcase
Z_{36}	DSAVM	- Available memory at end of subcase
Z_{37}	DSAVC	- Available communications at end of subcase
Z_{38}	CDISPR	- Dispersion at end of subcase (Number of nodes over which the application system is distributed)/(Number of application system modules)
Z_{39}	NCRIT	- Criticality of lost nodes (Sum of the connectivity of the application system modules residing on the lost nodes)/(Application system connectivity)

Z_{40}	DSCONN	- Distributed system connectivity
Z_{41}	MR/UMC	- Memory requirements/Useable memory capacity
Z_{42}	PR/UPC	- Processor requirements/Useable processing capacity
Z_{43}	CR/UCC	- Communications requirements/Useable communications capacity
Z_{44}	MINCUT	- NA
Z_{45}	ASCONN	- Application system connectivity
Z_{46}	DISPER	- Dispersion - initial (Number of nodes over which an application system is distributed/(Number of application system modules)
Z_{47}	MRCONS	- Memory consistency (Number of application system modules)/(Average number of application system modules that will "fit" on a node memory-wise)
Z_{48}	PRCONS	- Processor consistency (Number of application system modules)/(Average number of application system modules that will "fit" on a node processor-wise)
Z_{49}	LKCONS	- Link consistency (Average number of module to module interactions)/(capacity of a single link)
Z_{50}	DST	- NA
Z_{51}	DPOLICY	- NA

The factors which relate to dispersion are established to obtain insight into the implications of dispersion at various points in system operation. DISPER, X_{46}, represents initial dispersion. If n equals the number of nodes and m the number of modules the maximum initial

dispersion is one. That is every application module resides on a different node. The closer this ratio comes to 1/m the less dispersed the application system is said to be.

CDISPR, Z_{38}, has a similar interpretation, however, this measurement is taken at the end of the test case or after a certain percent of the nodes are lost.

NCRIT, Z_{39}, represents node criticality. This criticality is determined by summing the connectivity of the application system modules on the nodes which are lost and dividing this sum by the total application system connectivity. As this ratio approaches one, the proportion of the application system to be reallocated is increasing. Also, the character of the portion of the application system to be reallocated is described in terms of its need for cohesion.

The consistency measurements Z_{47}, Z_{48} describe the system in terms of memory and processing demands versus unit node memory and processing capacity. A ratio of one or less indicates that all memory or processing demands can be satisfied by a single node. Ratios greater than one indicate the number of nodes necessary to meet the demands. Note no consideration is made here concerning the capability of the system to make distributions which would use resourses optimally.

Link consistency, Z_{49}, varies slightly from the previous two consistency measures in that it relates average module to module interaction frequency to communication link capacity. This ratio indicates what portion of a link's capacity is consumed by average module to module interaction. The closer this ratio comes to one the

more likely modules will have to reside on the same node or have dedicated links.

Other data values derived were generated via transformations during data analysis. These values represent interactions among other variables. Six new variables of this type were created. These values are calculated by multiplication of the values of variables for which interaction is to be determined. They are

Z_{52} = Interaction among topologies

Z_{53} = $Z_2 \times Z_3 \times Z_4$
(Interaction between number of nodes and node processing speed and node memory capacity)

Z_{54} = $Z_6 \times Z_7 \times Z_8 \times Z_9$
(Interaction between number of application system modules and average module processing requirements and average module memory requirements and average module to module interaction frequency)

Z_{55} = $Z_2 \times Z_{18}$
(Interaction between number of nodes and number of lost nodes)

Z_{56} = $Z_{45} \times Z_{38}$
(Interaction between application system connectivity and dispersion at the end of a subcase)

Z_{57} = $Z_{45} \times Z_{38} \times Z_{40}$
(Interaction between application system

connectivity and dispersion at end of subcase and

distributed network connectivity)

Table 5.

```
SURVIVABILITY SIMULATOR

EXPERIMENT RUN DESCRIPTION

** CASE NUMBER   33  **

---------------------------------------------------------------------
                        ORIGINAL  Z-FACTORS
     RUN   1 2 3 4 5 6    7      8      9      10              11
---------------------------------------------------------------------

          - - - - - - + -   +      +      -      -      -      +      +


TYPE OF DISTRIBUTED SYSTEM TOPOLOGY:       STAR
NUMBER OF NODES:                              4

NODE PROCESSING SPEED:                      500 KOPS
NODE MEMORY CAPACITY:                       128 KBYTES

CONNECTIVITY OF APPLICATION SYSTEM:        HIGH
NUMBER OF APPLICATION MODULES FOR
     A GIVEN "SIZE" PROGRAM:                  4
AVERAGE MODULE PROCESSING REQUIREMENTS:       50% OF NODE PROCESSING SPEED
AVERAGE MODULE MEMORY REQUIREMENTS:         80.0% NODE MEMORY CAPACITY
AVERAGE FREQUENCY OF MODULE TO MODULE
     INTERACTION (# OF MESSAGE SETUPS):    LOW

DISTRIBUTION/REDISTRIBUTION POLICY:        RANDOM

NUMBER OF NODES ELIMINATED:                 90.0%
```

Table 6. Application System Topology Interaction Incidence Matrix

	A	B	C	D
A	0.00	0.63	0.13	0.05
B	0.18	0.00	0.19	0.37
C	0.20	0.41	0.00	0.15
D	0.01	0.42	0.74	0.00

Table 7. Application System Requirements

MODULE IDENTIFIER	MEMORY K BYTES	KOPS/ EXECUTION	EXECUTIONS /T	CRITICALITY
A	92.	100.	4.	2.
B	52.	188.	2.	3.
C	91.	139.	1.	1.
D	24.	63.	4.	4.

Table 8. Degradation Policy

STEP	
1	* Degrade modules of criticality equal 1 to .5 CPU executions memory communications
2	* Purge module D
3	* Degrade Modules of criticality less than 3 to .5 CPU executions memory communications
4	* Failure

Table 9. DISTRIBUTED SYSTEM TOPOLOGY INTERACTION INCIDENCE MATRIX

	1	2	3	4
1	100	100	0	0
2	100	100	100	100
3	0	100	100	0
4	0	100	0	100

QUEUE OF STARTNODES: 1, 2,

TABLE 10. DISTRIBUTED SYSTEM CAPABILITY

NODE	MEMORY K BYTES	CPU KOPS	# LINKS IN	CAPACITY IN	# LINKS OUT	CAPACITY OUT
1	128	500	1	100	1	100
2	128	500	3	300	3	300
3	128	500	1	100	1	100
4	128	500	1	100	1	100

Table 11. Resources Remaining After Initial Assignment

Initial Assignment was successful

Availability Matrix

NODE	Memory K bytes	CPU KOPS	# Links IN	Capacity IN	# Links OUT	Capacity OUT
1	36	100	1	100	1	100
2	76	124	3	300	3	300
3	128	500	1	100	1	100
4	13	109	1	100	1	100

Table 12. Module to Node Assignment

Node	Modules
1	A
2	B
3	
4	C D

Table 13. Sample Data Log

DATA1 -- 128 EXPERIMENT RUNS FOR 2(K-P) DESIGN

RUN	CASE	S	P	DIS	N	NPS	NNC	C	M	NPD	NNR	IF	PCL	B	SN	TAS	R	D	AS	W	NIW	GLP	GLM	CLC	AVE	AVM	AVC
1	1	0	4	1	4	500	128	1	4	50	80	2	4	10	1	2	0	0	0	2	0	2000.	512.	1000.	976.	287.	0.00
2	2	0	3	3	4	500	128	1	4	10	80	1	2	50	1	1	0	0	0	2	0	2000.	512.	1400.	1801.	204.	1399.81
3	3	1	1	2	4	500	128	1	4	10	10	1	4	30	1	1	0	0	0	2	0	2000.	512.	1200.	1723.	440.	1199.41
4	4	1	3	4	4	500	128	1	4	50	10	2	2	80	1	1	0	0	0	2	0	2000.	512.	1200.	860.	461.	1116.69
5	5	0	2	1	10	500	128	1	4	10	10	2	3	50	1	1	0	0	0	2	0	5000.	1280.	2800.	4723.	1229.	2800.00
6	6	1	1	3	10	500	128	1	4	50	10	1	1	10	1	1	0	0	0	2	0	5000.	1280.	3800.	3852.	1227.	3798.61
7	7	0	4	2	10	500	128	1	4	50	80	1	3	80	1	1	0	0	0	2	0	5000.	1280.	3000.	3783.	895.	2069.69
8	8	0	1	4	10	500	128	1	4	10	80	2	1	30	1	1	0	0	0	2	0	5000.	1280.	3600.	4832.	934.	3557.02
9	9	0	3	1	4	10000	128	1	4	50	10	1	3	30	1	1	0	0	0	2	0	40000.	512.	1000.	14053.	460.	997.46
10	10	1	1	3	4	10000	128	1	4	10	10	2	1	80	1	1	0	0	0	2	0	40000.	512.	1400.	36257.	450.	1341.22
11	11	0	4	2	4	10000	128	1	4	10	80	2	3	10	1	1	0	0	0	2	0	40000.	512.	1200.	37244.	151.	1192.97
12	12	0	4	4	4	10000	128	1	4	50	80	1	1	50	1	2	0	0	100	2	0	40000.	512.	1200.	35740.	307.	1200.00
13	13	0	3	1	10	10000	128	1	4	10	80	1	4	80	1	1	0	0	0	2	0	100000.	1280.	2800.	96916.	961.	2799.64
14	14	1	1	3	10	10000	128	1	4	50	80	2	2	30	1	1	0	0	0	2	0	100000.	1280.	3800.	75880.	908.	3799.16
15	15	0	4	2	10	10000	128	1	4	50	10	2	4	50	1	1	0	0	0	2	0	100000.	1280.	3000.	86026.	1230.	2866.91
16	16	1	3	4	10	10000	128	1	4	10	10	1	2	10	1	1	0	0	0	2	0	100000.	1280.	3600.	97172.	1223.	3600.00
17	17	1	3	1	4	500	2000	1	4	50	10	2	2	80	1	1	0	0	0	2	0	2000.	8000.	1000.	852.	2169.	610.42
18	18	1	1	3	4	500	2000	1	4	10	10	1	4	30	1	1	0	0	0	2	0	2000.	8000.	1400.	1494.	2135.	1399.30
19	19	0	3	2	4	500	2000	1	4	10	80	1	2	50	1	1	0	0	0	2	0	2000.	8000.	1200.	1917.	2074.	1199.86
20	20	0	4	4	4	500	2000	1	4	50	80	2	4	10	1	1	0	0	0	2	0	2000.	8000.	1200.	914.	3512.	1113.67
21	21	1	1	1	10	500	2000	1	4	10	80	2	1	30	1	1	0	0	0	2	0	5000.	20000.	2800.	4786.	16474.	2800.00
22	22	0	3	3	10	500	2000	1	4	50	80	1	3	80	1	1	0	0	0	2	0	5000.	20000.	3800.	4400.	15571.	3797.77
23	23	1	1	2	10	500	2000	1	4	50	10	1	3	10	1	1	0	0	0	2	0	5000.	20000.	3000.	4386.	19172.	2095.24
24	24	0	2	4	10	500	2000	1	4	10	10	2	3	50	1	1	0	0	0	2	0	5000.	20000.	3600.	4845.	19146.	3600.00
25	25	0	4	1	4	10000	2000	1	4	50	80	1	1	50	1	2	0	0	50	1	0	40000.	8000.	1000.	23028.	5839.	1000.00
26	26	0	3	3	4	10000	2000	1	4	10	80	2	3	10	1	1	0	0	0	2	0	40000.	8000.	1400.	37645.	2200.	1390.81
27	27	1	1	2	4	10000	2000	1	4	10	10	2	1	80	1	1	0	0	0	2	0	40000.	8000.	1200.	34608.	7182.	1155.48
28	28	0	3	4	4	10000	2000	1	4	50	10	1	3	30	1	1	0	0	0	2	0	40000.	8000.	1200.	14948.	7161.	1196.79
29	29	1	1	1	10	10000	2000	1	4	10	10	1	2	10	1	1	0	0	0	2	0	100000.	20000.	2800.	95238.	19161.	2800.00
30	30	0	4	3	10	10000	2000	1	4	50	10	2	4	50	1	1	0	0	0	2	0	100000.	20000.	3800.	82393.	19154.	3780.62
31	31	1	1	2	10	10000	2000	1	4	50	80	2	2	30	1	1	0	0	0	2	0	100000.	20000.	3000.	82834.	13692.	2969.26
32	32	0	4	4	10	10000	2000	1	4	10	80	1	4	80	1	1	0	0	0	2	0	100000.	20000.	3600.	95463.	15356.	3599.87
33	33	1	3	1	4	500	128	2	4	50	80	1	1	80	1	1	0	0	0	2	0	2000.	512.	1000.	833.	254.	997.81
34	34	0	3	3	4	500	128	2	4	10	80	2	3	30	1	1	0	0	0	2	0	2000.	512.	1400.	1657.	235.	1313.55
35	35	1	1	2	4	500	128	2	4	10	10	2	1	50	1	1	0	0	0	2	0	2000.	512.	1200.	1652.	463.	1200.00
36	36	0	4	4	4	500	128	2	4	50	10	1	3	10	1	1	0	0	0	2	0	2000.	512.	1200.	1047.	461.	1198.02
37	37	1	1	1	10	500	128	2	4	10	10	1	2	30	1	1	0	0	0	2	0	5000.	1280.	2800.	4708.	1225.	2800.00
38	38	0	4	3	10	500	128	2	4	50	10	2	4	80	1	1	0	0	0	2	0	5000.	1280.	3800.	4280.	1233.	3710.37
39	39	1	1	2	10	500	128	2	4	50	80	2	2	10	1	1	0	0	0	2	0	5000.	1280.	3000.	4220.	693.	2047.79
40	40	0	4	4	10	500	128	2	4	10	80	1	4	50	1	1	0	0	0	2	0	5000.	1280.	3600.	4826.	944.	3598.99
41	41	0	4	1	4	10000	128	2	4	50	10	2	2	50	1	2	0	0	0	2	0	40000.	512.	1000.	18412.	460.	0.00
42	42	0	1	3	4	10000	128	2	4	10	10	1	4	10	1	1	0	0	0	2	0	40000.	512.	1400.	35991.	466.	1399.29
43	43	0	4	2	4	10000	128	2	4	10	80	1	2	80	1	1	0	0	0	2	0	40000.	512.	1200.	34310.	179.	1199.04
44	44	0	3	4	4	10000	128	2	4	50	80	2	4	10	1	1	0	0	0	2	0	40000.	512.	1200.	14541.	163.	1024.29
45	45	1	1	1	10	10000	128	2	4	10	80	2	1	10	1	2	0	0	50	2	0	100000.	1280.	2800.	97425.	1133.	2800.00
46	46	0	3	3	10	10000	128	2	4	50	80	1	3	50	1	1	0	0	0	2	0	100000.	1280.	3800.	53673.	887.	3797.72
47	47	0	3	2	10	10000	128	2	4	50	10	1	1	30	1	2	0	0	30	1	0	100000.	1280.	3000.	81452.	1241.	3000.00
48	48	0	2	4	10	10000	128	2	4	10	10	2	3	80	1	1	0	0	0	2	0	100000.	1280.	3600.	95875.	1232.	3600.00
49	49	0	4	1	4	500	2000	2	4	50	10	1	3	10	1	1	0	0	0	2	0	2000.	8000.	1000.	1344.	2173.	493.02
50	50	1	1	3	4	500	2000	2	4	10	10	2	1	50	1	1	0	0	0	2	0	2000.	8000.	1400.	1802.	2198.	1377.83

Table 13 Continued. Sample Data Log

DATA1 -- 128 EXPERIMENT RUNS FOR 2(K-P) DESIGN

RUN	CASE	DSCONN	HWZLNC	PRZ/PC	CRZ/CC	MINCLT	ASCONN	DISPER	MRCCNS	PRCCNS	IXCCNS
1	1	0.500	0.440	0.510	0.240	1.000	0.500	1.000	3.200	2.000	0.250
2	2	0.830	0.600	0.100	0.300	1.000	0.500	1.000	3.200	0.400	0.000
3	3	0.670	0.100	0.140	0.000	1.000	0.500	1.000	0.400	0.400	0.000
4	4	0.470	0.110	0.570	0.120	1.000	0.500	1.000	0.400	2.000	0.250
5	5	0.200	0.040	0.060	0.020	1.000	0.500	0.250	0.400	0.400	0.050
6	6	0.310	0.040	0.230	0.000	1.000	0.500	1.000	0.400	2.000	0.250
7	7	0.220	0.300	0.240	0.000	1.000	0.500	1.000	3.200	2.000	0.000
8	8	0.290	0.270	0.030	0.010	1.000	0.500	1.000	3.200	0.400	0.050
9	9	0.500	0.100	0.650	0.100	1.000	0.500	1.000	0.400	2.000	0.000
10	10	0.430	0.100	0.090	0.100	1.000	0.500	0.500	0.400	0.400	0.050
11	11	0.470	0.310	0.070	0.010	1.000	0.500	0.250	3.200	0.400	0.050
12	12	0.620	0.240	0.260	0.000	1.000	0.500	1.000	3.200	2.000	0.000
13	13	0.200	0.240	0.070	0.010	1.000	0.500	1.000	3.200	0.400	0.000
14	14	0.310	0.290	0.240	0.040	1.000	0.500	1.000	3.200	2.000	0.000
15	15	0.220	0.130	0.140	0.060	1.000	0.500	1.000	0.400	2.000	0.250
16	16	0.290	0.470	0.330	0.000	1.000	0.500	1.000	0.400	0.400	0.250
17	17	[illegible]	0.100	0.570	0.190	1.000	0.500	1.000	0.400	0.400	0.250
18	18	0.330	[illegible]	[illegible]	[illegible]	1.000	0.500	1.000	3.200	0.400	0.000
19	19	0.620	0.810	0.290	0.100	1.000	0.500	1.000	3.200	0.400	0.000
20	20	0.670	0.560	0.540	0.110	1.000	0.500	1.000	3.200	2.000	0.250
21	21	0.270	0.190	0.140	0.120	1.000	0.500	0.750	3.200	0.400	0.050
22	22	0.310	0.220	0.120	0.000	1.000	0.500	0.750	3.200	2.000	0.000
23	23	0.220	0.030	0.120	[illegible]	[illegible]	0.500	0.500	0.400	2.000	0.000
24	24	0.290	0.040	0.030	0.110	[illegible]	0.500	0.250	0.400	0.400	0.000
25	25	[illegible]	[illegible]	0.690	[illegible]	[illegible]	0.500	1.000	3.200	2.000	0.050
26	26	0.430	0.230	0.050	0.010	[illegible]	0.500	1.000	3.200	0.400	0.050
27	27	0.670	0.100	0.110	0.270	1.000	0.500	0.750	0.400	0.400	0.050
28	28	0.670	0.100	0.430	[illegible]	1.000	0.500	0.750	0.400	2.000	0.050
29	29	0.200	0.040	0.080	0.000	1.000	0.500	1.000	0.400	0.400	0.000
30	30	0.310	0.140	0.190	0.140	1.000	0.500	1.000	0.400	2.000	0.250
31	31	0.220	0.120	0.170	0.360	1.000	0.500	1.000	3.200	2.000	0.250
32	32	0.290	0.230	0.120	0.300	0.300	0.500	1.000	3.200	0.400	0.250
33	33	0.500	0.510	0.590	0.380	0.000	0.750	1.000	3.200	0.400	0.000
34	34	0.830	0.540	0.070	0.370	0.000	1.000	0.750	3.200	0.400	0.050
35	35	0.670	0.100	0.070	0.260	0.000	1.000	0.500	0.400	0.400	0.050
36	36	0.670	0.100	0.480	0.310	0.000	1.000	0.750	0.400	2.000	0.050
37	37	0.200	0.040	0.060	0.020	0.000	1.000	1.000	0.400	0.400	0.000
38	38	0.310	0.040	0.060	0.020	0.000	1.000	1.000	0.400	2.000	0.250
39	39	0.220	0.300	0.140	0.110	0.000	1.000	1.000	3.200	2.000	0.250
40	40	0.290	0.240	0.010	0.000	0.000	1.000	1.000	3.200	0.400	0.000
41	41	0.500	0.100	0.540	0.360	0.000	1.000	1.000	0.400	2.000	0.250
42	42	0.830	0.600	0.100	0.000	0.000	1.000	1.000	0.400	0.400	0.000
43	43	0.670	0.440	0.100	0.000	0.000	1.000	1.000	3.200	0.400	0.000
44	44	0.470	0.110	0.640	0.130	0.000	1.000	1.000	3.200	2.000	0.250
45	45	0.200	0.260	0.060	0.130	0.000	1.000	0.500	3.200	0.400	0.250
46	46	0.310	0.310	0.180	0.000	0.000	1.000	1.000	3.200	2.000	0.000
47	47	0.220	0.180	0.030	0.000	0.000	1.000	0.750	0.400	2.000	0.000
48	48	0.290	0.180	0.040	0.030	0.100	1.000	0.250	0.400	0.400	0.050
49	49	0.500	0.100	0.330	0.010	0.000	1.000	0.500	0.400	2.000	0.050
50	50	0.830	0.100	0.100	0.040	0.000	1.000	0.750	0.400	0.400	0.050

Table 13 continued. Sample Data Log

DATA3 -- 128 EXPERIMENT RUNS FOR 2(K-P) DESIGN

RUN	CASE	S	P	DSAVP	DSAVM	PSAVC	CDISPR	PCRIT
1	1	0.00	4.00	0.	0.	0.00	0.00	0.00
2	2	0.92	3.08	931.	32.	416.56	0.66	0.84
3	3	1.00	1.00	1723.	332.	609.61	0.75	0.50
4	4	1.00	3.00	19.	95.	100.00	0.33	1.00
5	5	0.55	2.36	2609.	698.	1410.61	0.25	0.45
6	6	1.00	1.08	3362.	1099.	3138.95	0.90	0.19
7	7	0.00	4.00	0.	0.	0.00	0.00	0.00
8	8	0.99	1.72	3491.	606.	2038.63	0.98	0.49
9	9	0.63	3.38	15178.	364.	686.56	0.62	0.50
10	10	1.00	1.00	6257.	75.	100.00	0.25	0.96
11	11	0.00	4.00	0.	0.	0.00	0.00	0.00
12	12	0.00	4.00	19573.	18.	698.99	0.67	0.72
13	13	0.18	3.82	37706.	257.	1005.30	0.95	0.90
14	14	1.00	1.00	48607.	559.	2112.56	1.00	0.49
15	15	0.00	4.00	0.	0.	0.00	0.00	0.00
16	16	1.00	1.00	87172.	1095.	2979.81	1.00	0.20
17	17	1.00	3.00	19.	1497.	100.00	0.33	1.00
18	18	1.00	1.00	1394.	5135.	799.48	0.75	0.50
19	19	0.67	1.33	953.	584.	383.21	0.64	0.84
20	20	0.00	4.00	0.	0.	0.00	0.00	0.00
21	21	1.00	1.15	3435.	11067.	1630.17	0.76	0.42
22	22	0.11	1.90	1620.	4663.	1106.35	0.74	0.87
23	23	1.00	1.00	3886.	17172.	2499.39	0.50	0.13
24	24	0.55	2.35	2647.	10353.	1451.17	0.25	0.45
25	25	0.00	4.00	23028.	5839.	1000.00	0.50	0.00
26	26	0.00	4.00	0.	0.	0.00	0.00	0.00
27	27	1.00	1.00	4808.	1182.	100.00	0.25	0.92
28	28	0.38	3.13	10373.	5979.	885.49	0.66	0.50
29	29	1.00	1.00	85238.	17161.	2339.95	1.00	0.20
30	30	0.00	4.00	0.	0.	0.00	0.00	0.00
31	31	1.00	1.00	55561.	8237.	1712.34	1.00	0.49
32	32	0.00	4.00	0.	0.	0.00	0.00	0.00
33	33	1.00	3.00	15.	7.	100.00	0.31	0.96
34	34	0.44	3.06	1495.	155.	848.14	0.67	0.46
35	35	1.00	1.00	852.	207.	314.47	0.50	0.67
36	36	0.00	4.00	0.	0.	0.00	0.00	0.00
37	37	1.00	1.00	3344.	876.	1643.33	1.00	0.49
38	38	0.00	4.00	0.	0.	0.00	0.00	0.00
39	39	1.00	1.00	3720.	775.	2458.23	1.00	0.20
40	40	0.00	4.00	0.	0.	0.00	0.00	0.00
41	41	0.00	4.00	0.	0.	0.00	0.00	0.00
42	42	0.00	4.00	0.	0.	0.00	0.00	0.00
43	43	0.00	4.00	16560.	45.	399.36	1.00	1.00
44	44	0.88	3.13	14475.	183.	685.79	0.95	0.50
45	45	1.00	1.00	85213.	818.	2298.88	1.00	0.20
46	46	0.12	3.79	43692.	389.	1731.33	1.00	0.73
47	47	0.00	4.00	81452.	1241.	3000.00	0.75	0.00
48	48	0.35	2.94	34560.	447.	883.42	0.75	0.65
49	49	0.00	4.00	0.	0.	0.00	0.00	0.00
50	50	1.00	1.00	402.	3198.	350.23	0.44	0.78

CHAPTER VI

ANALYSIS PART I

Introduction to Regression

Regression is a technique used to quantify the relationship between variables when the value of one variable, called the response or *dependent* variable, is affected by changes in the values of other variables, called predictor or independent variables. The correlation between any two variables is often used to indicate whether an increase or decrease in one is associated with a corresponding increase or decrease in the other. The relationship between a response variable, y, and an independent variable, x, is said to be linear if the expected value of y, usually stated E(y), can be expressed in the form

$$E(y) = \alpha + \beta X \tag{6-1}$$

Here α and β are parameters of a regression equation in which α represents the intercept and β the slope of the regression. Where there is only one independent variable as in this example the regression is termed simple linear regression. The experiments conducted in this research involved a large number of independent variables. The regression equation in this case appears as follows

$$E(y) = \alpha + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_K X_K + \epsilon \tag{6-2}$$

and is *termed* a *multiple* linear regression model. Given a model of

this form, selection of variables is very important. When, as in the case of this research, data has been collected on more variables than may be necessary in the model, established variable selection techniques can be employed to assist in deciding the most suitable variable mix to use in the final model. The consequences of poor variable selection fall into two general categories. The final model may be 1.) useless or misleading because an important variable has been omitted or 2.) unused because the inclusion of extraneous variables has caused it to be cumbersome. Since no single approach to variable selection is guaranteed to produce the "best" model, it is common to utilize several variable selection techniques to aid in the model building process. The following three subsections describe the techniques used in this work. These techniques differ in 1.) the criterion used for selection of independent variables, 2.) the amount of analysis and comparison that is done using subgroups of the independent variables and 3.) the type of residual analysis performed. Residual here refers to the difference between the observed and model-generated or fitted values of a response variable.

Multiple Linear Regression

The Multiple Linear Regression technique for variable selection estimates the multiple linear regression equation using all of the independent variables. The coefficients of the regression model are estimated by least squares. This technique is performed by a computerized statistical package PMDP-Routine PIR. Output from this program includes a variety of standard statistical measures for each of the variables in the model. One of the measures provided is the T

statistic. The availability of this statistic for all candidate variables facilitates use of a variable selection approach called the directed search on T. The test statistic T will be large for those regressors which contribute significantly to the full model. If these regressors are introduced to the model one at a time in order of descending T value, the model should be at any given point the "best" or one of the best for that size subset of all possible regressors. The directed search on T is a good variable selection strategy when the number of variables is large, say 20 to 30.

Stepwise Regression

The stepwise regression technique enters and removes variables from a multiple linear regression equation in a stepwise manner. At each step in the model building process variables are removed and/or entered into the equation. The criteria for determining entry or removal of a variable is normally its F statistic when considered along with other variables in the model. Forward stepping is an approach which begins with no predictors and consecutively adds variables which exceed some threshold value. Backward stepping is an approach which begins with all candidate predictors and consecutively removes variables which fall below a given lower bound. Techniques used in this research employ a combination of forward selection and backward elimination.

All Possible Subsets Regression

The all possible subsets regression procedure requires the fitting of all of the regression equations involving one through n candidate regressors, where n is the number of variables. The number

of equations to be examined increases exponentially with the number of candidate regressor variables. To evaluate subset regression equations several measures can be used. These include R^2, coefficient of multiple determination; adjusted R^2, minimal residual mean square; MS_E, mean square for error; and Mallows C_p, residual sum of squares. All possible subsets regression using adjusted R^2 and Mallows Cp are used here.

In the adjusted R^2 evaluation, the T statistic for the coefficients of variables in the subset that maximizes adjusted R^2 are all greater than one in absolute value. Maximizing adjusted R^2 is the same as minimizing the residual mean square. Usually, subsets larger than those that maximize adjusted R^2 are not very good. In the Mallows Cp evaluation, the T statistic for the coefficients of variables in the subset that minimizes Cp are usually greater than $\sqrt{2}$ in absolute value. When using all possible subsets regression the problems of variable selection increase as the number of redundant variables increases. Inclusion of irrelevant variables provides the opportunity for artifacts in the data to produce unpredictably high T statistics, R^2, and adjusted R^2 and unpredictably low Mallows Cp statistics. For this reason checks must be made for variable redundancy and redundant variables removed from the set of candidate variables.

Procedures for Model Building

Data Reduction

The experimental design presented in Chapter IV described the 128 experiment runs necessary for a 2_V^{K-P} design in 14 factors. During execution of the simulator data was collected for each of these cases

plus two types of subcases. The subcases were those that tracked operational data for all possible unique start nodes and all possible number of nodes lost. The total number of cases and subcases logged by the simulator are in excess of 300,000. Physically this translates to approximately 90 megabytes of data which is one very large magnetic disk or seven 2,400 foot 1,600 BPI magnetic tapes. The mechanical difficulty of working with this volume of data suggests that the possibility of meaningful data reduction should be explored. Fortunately, some reduction of the data could be performed without significantly decreasing its information value. Therefore, before analysis the raw data was put through a data reduction filter which produced three sets of data, each of different resolution.

DATA 1 - comprises the 128 designed experiment runs times an averaging over all possible start nodes times an averaging over number of nodes lost. The size of this data set is 2,156 cases.

DATA 2 - comprises the 128 designed experiment runs times an averaging over number of nodes lost. The size of this data set is 715 cases.

DATE 3 - comprises the 128 designed experiment runs. The size of this data set is 128 cases.

DATA 1 has, of course, the highest resolution of the 3 data sets. It presents a summary of individual subcases such that specific information is lost concerning individual subcases for each start node

and each possible number of nodes lost. DATA 2 presents a summary which ignors the start node and DATA 3 presents a summary which ignors both start node and specific number of lost nodes. DATA 3 refers to node loss as a percent of the total nodes initially in the distributed system. Examination of the analyses conducted using all three data sets revealed that the designed data set, DATA 3, was representative of the other two.

Candidate Variables

The variables submitted to data analysis are of three types: response variables, control variables, and other independent variables. The response variables are survivability, S, and performance, P. Since S can be obtained from a simple function on P, the focus of discussion of analyses performed will be on P. The control variables are the 11 factors described in Chapter IV section 2. Other independent variables or potential control variables are those listed in Chapter V section 2.

Independent variables fall into two general categories: quantitative or continuous valued variables and indicator variables. Most often variables used in regression model building are quantitative or continuous valued variables which take on values within some known range on a well-defined scale. Less frequently, it is necessary to include qualitative variables which have no natural scale of measurement in the regression model. Qualitative variables, often represented as indicator or "dummy" variables are assigned a set of levels to account for the effect that the variable may have on the response. In this research all independent variables are quantitative with the exception of two. These are distributed system topology and

distribution policy. Both of these variables have four levels and thus require three "dummy" variables to represent them. This is accomplished by arbitrarily assigning one of the following codes to each of the qualitative variables.

DISTRIBUTED SYSTEM TOPOLOGY	"DUMMY" VARIABLE IA	IB	IC
STAR	0	0	0
RING	1	0	0
NETWORK	0	1	0
ARRAY	0	0	1

DISTRIBUTION POLICY	"DUMMY" VARIABLE ID	IE	IF
RANDOM	0	0	0
UNIFORM	1	0	0
PACKED	0	1	0
OPTIMAL SPARE	0	0	1

The interpretation given to coefficients of qualitative variables is different than that of quanitative variables in that the coefficient of a qualitative variable indicates the relative impact of change from that level to other possible levels of the qualitative variable. For example, in the case of topology each of the "dummy" variables when present in the model indicate the effect of change from the base level or condition, 000, to that level. The effect of change from one of the other levels to a third level is accomplished by subtracting the coefficients of the variables in question. Thus the effect of a change

DISTRIBUTED SYSTEM TOPOLOGY	"DUMMY" VARIABLE IA	IB	IC
STAR	0	0	0
RING	1	0	0
NETWORK	0	1	0
ARRAY	0	0	1

from the star to the ring topology is provided by the coefficient on IA, the star to the network by the coefficient on IB, etc. The effect of a change from the ring to the network is determined by subtracting the coefficient of IB from that of IA. The same approach is used for all comparisons. In the case of quantitative variables, the coefficients indicate the direction and magnitude of the relationship between the independent variable and the response.

An important part of regression analysis is variable selection. In the case of distributed processing systems the most appropriate set of regressor variables is not known and little prior experience exists which might help point the way to initial selection. In such cases it is desirable to begin with the most comprehensive set of candidate variables and reduce this number through iterative selection of regressor sets which are "best" according to one of the evaluation criteria listed above.

Explanatory versus Predictive Models

Usually, regression models are valid only over the range of the regressor variables contained in the observed data. Over this interval, the regression equation developed may provide a reasonable approximation of the true functional relationship. However, care

should be exercised to assure that the application of a regression model does not exceed its capability. For example, while some regression models may adequately summarize or describe the data from which they were constructed, they may be less serviceable in describing new data. A model which describes the data to which it was fit is called an explanatory model. Measurements can be made which indicate the adquacy of an explanatory model in fitting its data. Checking for explanatory model adequacy can be done via residual analysis, testing for lack of fit, searching for high-leverage or overly-influential observations and a variety of internal consistency checks (20). It should not be assumed that a model which is proved to fit existing data will also be a good predictor for future data. Further, the model that provides the best fit to existing data may not be equally successful in the final application, that is be a successful predicator. To determine how well the explanatory model will serve as a predictor requires that we validate the model. A number of techniques are available for model validation. These include comparison with other results, collection and comparison with new data, and data splitting. The approaches used to develop explanatory and prediction models for operational survivability are presented in the following sections.

The Explanatory Model Building Process

Building a regression model is generally an iterative process requiring repeated analyses as improvements in the model structure or additional special features of the data are discovered. Digital computers and established statistical software can be invaluable model building tools. In this case several regression routines comprised in

the BMDP statistical software package are used. These are P1R; Multiple Linear Regression; P2R, Stepwise Regression; and P9R, All Possible Subset Regression.

Initially multiple linear regression is performed using all candidate independent variables. A check is made for multicollinearity among the independent variables. Redundant variables identified by this check are removed from the list of candidate regressors, and the analysis is repeated. The model resulting from this analysis is shown in Figure 3. Also provided in Table 14 are major statistics such as R^2 and Mean Square for Error for the model and T-statistic, mean and standard deviation for each of the regressor variables.

NOTE: See Table 16 for Variable Key

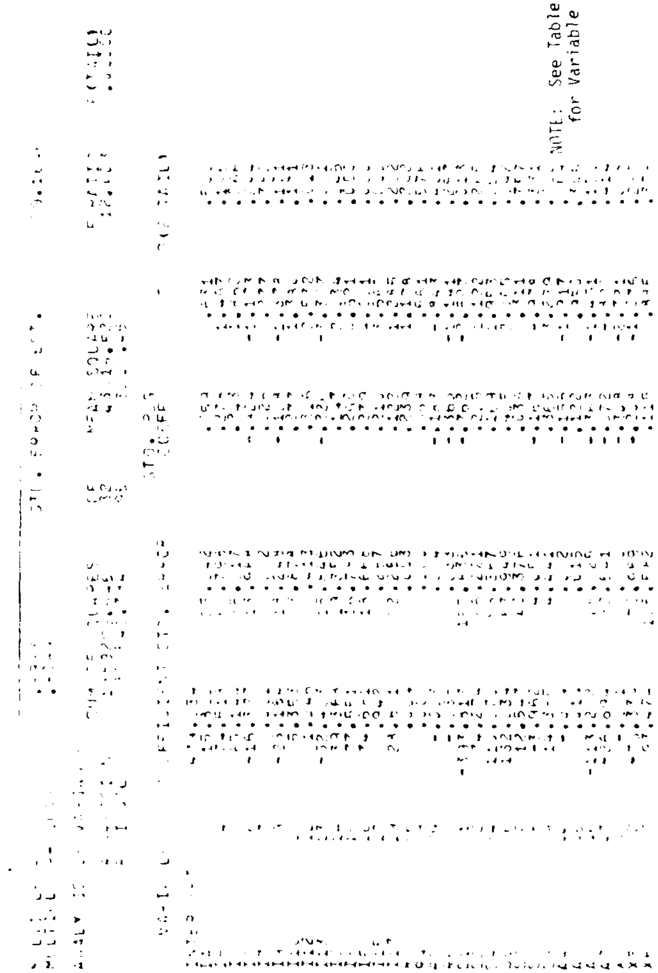


Figure 3. Multiple Linear Regression Model with all Candidate Regressor Variables

Table 14. Statistics from BMDP Multiple Linear Regression Analysis

Variable	Mean	Standard Deviation	St. Dev. Mean	Minimum	Maximum
X1	.25000	.43471	1.73886	0.00000	1.00000
X2	.25000	.43471	1.73886	0.00000	1.00000
X3	.25000	.43471	1.73886	0.00000	1.00000
X4	7.00000	3.01179	.43026	4.00000	10.00000
X5	250.00000	4768.66412	.90832	500.00000	10000.00000
X6	1.50000	.50196	.33464	1.00000	2.00000
X7	10.00000	6.02358	.60236	4.00000	16.00000
X8	45.00000	35.13753	.78083	10.00000	80.00000
X9	1.50000	.50196	.33464	1.00000	2.00000
X10	.25000	.43471	1.73886	0.00000	1.00000
X11	.25000	.43471	1.73886	0.00000	1.00000
X12	.25000	.43471	1.73886	0.00000	1.00000
X13	42.50000	25.96181	.61087	10.00000	80.00000
X14	1.43750	.49803	.34645	1.00000	2.00000
X15	7448.00000	7841.09470	1.05278	512.00000	20000.00000
X16	27896.63281	33402.45299	1.20059	331.00000	97425.00000
X17	5827.65625	6605.86220	1.13354	137.00000	19304.00000
X18	2217.51047	1125.45843	.50753	0.00000	3800.00000
X19	.46125	.22541	.48869	.20000	.83000
X20	.35594	.32141	.90300	.04000	1.05000
X21	.09312	.17906	1.92284	0.00000	1.05000
X22	.44250	.35695	.80667	.10000	1.00000
X23	.59797	.32887	.54997	.06000	1.00000
X24	4.50000	4.91310	1.09180	.40000	12.80000
X25	3.00000	2.95776	.98592	.40000	8.00000
X26	1997.70312	4095.96502	2.05034	0.00000	17172.00000
X27	794.59961	1007.87230	1.43042	0.00000	3600.00000
X28	.29836	.37377	1.25273	0.00000	1.00000
X29	.21992	.31591	1.43648	0.00000	1.00000
X30	20250.00000	32163.95008	1.58834	400.00000	128000.00000
X31	.20459	.31684	1.54874	0.00000	1.00000
X32	.08424	.14263	1.69324	0.00000	.67000

NOTE: See Table 16 for Variable Key

Next, stepwise is performed. This analysis provides an incremental view of the model as it is being developed. The point at which model building using stepwise regression can be considered complete is at the point in which the R^2 value begins to show only nominal increases and the mean square for error, MS_E, starts to increase. Figure 4 provides a picture of the regression model at this point. A quick validation can be made at this stage. To perform this validation a directed search on T for the results of PIR must be conducted. This search constitutes a ranking of candidate regressor variables according to descending values of T. When this list is compared to the list of regressor variables proposed as a result of *stepwise regression analysis*, the variables with the largest T statistic after the direct search on T should roughly correspond to the variables remaining in the model after stepwise analysis.

NOTE: See Table 16 for variable key

Figure 4 Stepwise Regression Model

The two analyses conducted at this point should serve to reduce the number of candidate regressor variables. All possible subsets regression (P9R) is now performed using the remaining variables. This model building technique is executed first using Mallows Cp as the variable selection criterion, *then using* adjusted R^2 as the variable selection criterion. Each of these provides detailed information on the five subset models determined to be "best" according to the evaluation criterion in effect *and the one model* considered optimum. The five models developed using Mallows Cp and adjusted R^2 as evaluation criteria are contained in Appendix B. The two optimum models are presented in Figures 5 *and* 6 respectively. Table 15 compares the models developed by multiple linear regression, stepwise regression, and all possible susbsets regression according to the evaluation data available.

STATISTICS FOR [illegible]
MALLOWS [illegible]
SQUARE [illegible] MULTIPLE CORRELATION [illegible]
MULTIPLE CORRELATION [illegible]
ADJUSTED [illegible]
RESIDUAL [illegible]
STANDARD ERROR OF [illegible]
F-STATISTIC [illegible]
NUMERATOR [illegible]
DENOMINATOR [illegible] OF FREEDOM [illegible]
SIGNIFICANCE [illegible]

VARIABLE NO. NAME	[illegible] COEFFICIENT	STANDARD ERROR	STAND. COEF.	T- STAT.	2TAIL SIG.	TOL- ERANCE	CONTRIBUTION TO R-SQUARED
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

NOTE See Table [illegible] for variable key
THE CONTRIBUTION TO [illegible] FOR [illegible] VARIABLE IS THE AMOUNT
BY WHICH [illegible] IF THAT VARIABLE WERE
REMOVED [illegible]

Figure 5 Optimum Model According to Mallows Cp Criterion

STATISTICS [illegible]
[illegible]

VARIABLE NO. NAME	[illegible] COEFFICIENT	STANDARD ERROR	STAND. COEF.	T- STAT.	2TAIL SIG.	TOL- ERANCE	CONTRIBUTION TO R-SQUARED
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

NOTE See Table [illegible] for variable key
THE CONTRIBUTION [illegible] VARIABLE IS THE AMOUNT
[illegible] THAT VARIABLE WERE
[illegible]

Figure 6 Optimum Model According to Adjusted R Criterion

Table 15. Comparison of Models Constructed by Three Regression Methods

Method/Model	Number of Variables	R^2	MS_E	DF	Adjusted R^2	Mallows Cp
Multiple Linear Regression	32	.8069	3500.4	95	-	-
Stepwise Regression	9	.7392	380618.8	118	-	-
All Possible Subsets Regression - A						
1	15	.765890	-	-	.734536	15.03
2	16	.769484	-	-	.736256	15.32
3	18	.777987	-	-	.741325	15.29
4	19	.782770	-	-	.744554	15.02
5	19	.782582	-	-	.744332	15.11
All Possible Subsets Regression - B						
1	24	.797205	-	-	.749952	18.17
2	24	.796944	-	-	.749630	18.30
3	24	.796902	-	-	.749579	18.32
4	25	.799129	-	-	.749896	19.26
5	26	.801106	-	-	.749905	20.32

R^2 - R-Squared
MS_E - Mean Square for Error - Residual
DF_E - Degree of Freedom - Residual

Prediction Model Building Process

There are a number of ways in which regression models can be validated and their value as predictors evaluated. Methods of model validation fall into three general categories. These are:

1) analysis of model coefficients and predicted values including comparisons with prior experience, physical theory, other analytical models or simulation results,
2) collection of fresh data with which to investigate the models predictive performance,
3) data splitting; breaking the original data into groups and using these observations to predict the model's performance as a predictor.

Data splitting, which is the approach taken here, is accomplished by separating available data into two parts, the estimation data and the prediction data. The estimation data is used to build the regression model. The prediction data is then used to study the predictive ability of the model. This technique is also called cross-validation.

Since the experiment conducted for this reasearch is a "designed" experiment, data splitting can be accomplished in a very straightforward manner. One of the factors in the original eleven that will not be included in the final model is used as the determinant for the split. The variable to be used is Z_4, absolute memory size.

Based on this factor, the data is split into two groups, let us call them DATA A and DATA B. Using DATA B as the estimation data set, new models are fit using only the variables specified for the optimal

models designated by the all possible subsets regression. A total of 32 variables are possible in the new models. The descriptions for these variables as they are renamed are given in Table 16. A chart showing which variables are used in which models is provided in Table 17. The 10 new models fit using multiple linear regression on DATA B, the estimation set, are presented in Appendix C. Each of these models is used to predict the response values of DATA A, the prediction set. The adequacy of the fitted models as predicators is determined by an R^2 for prediction computed as follows.

$$R^2 \text{ prediction} = 1 - \frac{\sum_{i=1}^{N} e_i^2}{S_{yy}} \qquad (6\text{-}3)$$

where $e = y - \hat{y}$

in which y is the observed value of the response

$\hat{y}$ is the fitted value of the response

and

$$Syy = \sum_{i=1}^{N} (y_i - \bar{y})^2 \qquad (6\text{-}4)$$

is the corrected sum of squares in which $\bar{y}$ is the mean of the observed responses

The R^2 prediction can occasionally be modified upward in instances when the fitted values of the response exceed the range of the observed response only by a small percent. This modification introduces at the model prediction evaluation phase the same correction that would otherwise be made when using the model as a predicator.

Table 18 is a comparison of the 10 fitted models in terms of their explanatory and predictive performance.

Table 16. Candidate Regressors - Variable Key

Factor Label Code/Variable		Factor Description
IA	X1	Dummy Variables Indicating
IB	X2	Distributed System Topology
IC	X3	
I6	X4	Number of Nodes in the Distributed System
I7	X5	Node Processing Speed
I9	X6	Node Communications Capacity
I10	X7	Number of Application System Modules
I12	X8	Module Memory Requirements
I13	X9	Module to Module Interaction Frequency
ID	X10	Dummy Variables Indicating Distribution Policy
IE	X11	
IF	X12	
I15	X13	Percent Nodes Lost
I17	X14	Initial Assignment Result
R2	X15	Global Memory Capacity
R4	X16	Available Processing Capacity after Initial Assignment
R5	X17	Available Memory Capacity after Initial Assignment
R6	X18	Available Communications Capacity after Initial Assignment
S1	X19	Distributed System Connectivity
S2	X20	Memory Requirements/Useable Memory Capacity
S4	X21	Communications Requirements/Useable Communications Capacity
S6	X22	Application System Connectivity
S7	X23	Dispersion - Initial (Number of nodes over which an application system is distributed/ (Number of application system modules)
S8	X24	Memory Consistency (Number of application system modules)/ (Average number of application system modules that will "fit" on a node - memory wise)

Table 16 continued. Candidate Regressors

Factor Label Code/Variabl		Factor Description
S9	X25	Processor Consistency (Number of application system modules)/ (Average number of application system modules that will "fit" on a node-processorwise)
A4	X26	Available Processing Capacity at End of Subcase
A5	X27	Available Communications Capacity at End of Subcase
A6	X28	Dispersion at End of Subcase (Number of nodes over which the application system is distributed)/ (number of application system modules)
A7	X29	Criticality of Lost Nodes (Sum of the connectivity of the application system modules residing on the lost nodes)/ (Application system connectivity)
X3	X30	Interaction between Number of Application System Modules and Module Processing Requirements and Module Memory Requirements and Module Communication Requirements
X5	X31	Interaction between Dispersion at End of Subcase and Application System Connectivity
X6	X32	Interaction between Dispersion at End of Subcase and Application System Connectivity and Distributed System Connectivity

Table 17. X Factors Present in 10 Best Subsets Models

MODEL	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}	X_{11}	X_{12}	X_{13}	X_{14}	X_{15}	X_{16}
1				X				X	X	X	X	X	X	X		
2				X	X			X	X	X	X	X	X	X		
3				X	X			X	X	X	X	X	X	X		
4	X			X	X			X	X	X	X	X	X	X		
5	X			X				X	X	X	X	X	X	X		X
6		X	X	X		X	X	X	X	X	X	X	X	X	X	X
7		X	X	X	X			X	X	X	X	X	X	X	X	
8		X	X	X	X	X	X	X	X	X	X	X	X	X	X	
9		X	X	X		X	X	X	X	X	X	X	X	X	X	X
10	X	X	X	X		X	X	X	X	X	X	X	X	X	X	X

MODEL	X_{17}	X_{18}	X_{19}	X_{20}	X_{21}	X_{22}	X_{23}	X_{24}	X_{25}	X_{26}	X_{27}	X_{28}	X_{29}	X_{30}	X_{31}	X_{32}
1			X			X			X	X		X	X	X		
2			X			X			X	X		X	X	X		
3			X		X		X		X	X	X	X		X		X
4			X		X		X		X	X	X	X		X		X
5			X		X		X		X	X	X	X		X		X
6			X	X	X	X		X	X	X			X	X	X	
7	X		X		X	X			X	X	X	X	X	X	X	X
8			X	X	X	X		X	X	X			X	X	X	
9			X	X	X	X	X	X	X	X	X	X		X		
10		X	X	X	X	X		X	X	X			X	X	X	

NOTE: See Table 16 for Variable Key

Table 18. Comparison of Model Adequacy

MODEL NUMBER	EXPLANATORY R^2	PREDICTION R^2	PREDICTION TRIM R^2
1	.8317	.52125	.53577
2	.8326	.65498	.70286
3	.8457	.38786	.47850
4	.8559	.19704	.33184
5	.8551	.20360	.33173
6	.8622	.65325	.71536
7	.8652	-.39189	.05569
8	.8626	.64696	.71252
9	.8647	.41798	.50232
10	.8641	.50405	.66264

Another method of prediction validation is to reverse the roles of the prediction and estimation sets and check the results for consistency. DATA A then becomes the estimation set and DATA B the prediction set. Multiple linear regression is used to fit models for the estimation set. These models are in turn used to predict observed values in DATA B. The explanatory R^2 values for the models built on DATA A are consistently lower than those built on DATA B. In addition when the R^2 for prediction was assessed, only one of the fitted models proved to be a good predictor. That model, number 10, had the largest number of regressors and an R^2 prediction of .65958. The remaining nine models made almost poor predictions. Upon cross examination of the data in the sets DATA A and DATA B after the split, it was determined that the two sets were equivalent in all respects with the exception of three variables. These three variables were indirectly related to the variable which formed the basis for the split. This relationship, thus, caused the high values of these three variables to be in one set and the low value in the other. These three variables X_{15}, X_{17}, and X_{26} were indirectly related to the response and were present either individually or in groups in most of the models. Fitted models built on the data set with the low values of these variables were for the most part unstable when used as predictors. When the fitted models were built on the data set with the high values the models were stable, however, consistently underpredicted the performance of the data set having the low values. No correction was made for this underprediction because the adjustment would be unique to predicting into DATA A. It is believed that a more representative

prediction set would reveal a stronger prediction capability than is indicated here. A split of the data such that DATA A and DATA B are equivalent on all variables may be possible and should substantiate further the findings presented here.

CHAPTER VII

ANALYSIS PART II - INTERPRETATION

Discussion of Explanatory Prediction and Models

In the 10 best subset models resulting from all possible subset regression analysis, a total of 32 candidate regressor variables are possible. The variables included in each of these subset models is presented in Table 17 and a description of each of the variables is provided in Table 16. Certain variables are found in all 10 models. These are X_4, X_8, X_9, X_{10}, X_{11}, X_{12}, X_{13}, X_{14}, X_{19}, X_{26}, and X_{30}, which represent number of nodes in the distributed system, module memory requirements, module to module interaction frequency, distribution policy, percent nodes lost, initial assignment result, distributed system connectivity available processing capacity at the end of the subcase and the interaction of all application system related variables. Table 19 presents the coefficients for the variables in the 10 best subset models. As can be observed in this table the coefficients for the nine variables found in all 10 models are approximately equivalent in sign and magnitude for all models. In fact, there exists extreme stability of all coefficients across models. Changes when they occur are proportional. Equivalent signs and magnitudes means that the regression coefficients are good estimates of the effects of these factors upon performance. Also, these variables form the core of a model that will likely be good for both explanatory and predictive assessment. In other words, the 32 variables included

in these models are very stable and are not distorted much by the introduction or removal of other variables.

The number of variables in addition to the nine foundation variables needed to achieve explanatory models with R^2 adequacy levels above .8 vary between four and 11. It is important to note that among the nine essential factors are factors which represent each of the three categories hypothesized at the outset of this research. That is X_4 and X_{26} pertain to the distributed system network; X_8, X_9, X_{30} pertain to the application system; and X_{10}, X_{11}, X_{12}, and X_{14} pertain to the distribution policy.

The interpretation of coefficients describing the influence of qualitative variables is different than the interpretation of coefficients of quantitative variables. The coefficient of qualitative or indicator variables such as X_1, X_2, X_3 and X_{10}, X_{11}, X_{12} describe the impact of change to that level from another level. The interpretation of coefficients modifying quantitive variables is traditional. That is, a positive coefficient corresponds to a direct relationship with the response variable and a negative coefficient designates an inverse relationship. It should be pointed out, however, in this research that the higher the value of the response variable the worse the performance. A strong inverse relationship between the variable and response is "good." Considerable caution should still be exercised in interpreting regression coefficients because regression does not imply causality. That is, there may be a strong correlative relationship between the factors which results in a significant regression, but the factors may not be related in a cause and effect fashion (20).

Table 19. Coefficients for Variables in 10 Best Subsets Models

VARIABLE X ( )	COEFFICIENTS MODEL - 1	MODEL - 2	MODEL - 3	MODEL - 4	MODEL - 5
1	-	-	-	-29.706	-29.934
2	-	-	-	-	-
3	-	-	-	-	-
4	-20.793	-20.783	-22.407	-23.658	-24.001
5	-	0.001	0.001	0.001	-
6	-	-	-	-	-
7	-	-	-	-	-
8	1.149	1.149	0.982	0.865	0.861
9	-23.276	-23.146	-50.523	-53.975	-53.780
10	46.484	46.653	22.008	12.177	11.921
11	76.547	77.101	75.362	74.753	74.570
12	77.873	78.371	60.792	53.856	53.498
13	0.686	0.686	0.624	0.611	0.612
14	37.917	39.473	41.099	47.884	47.315
15	-	-	-	-	-
16	-	-	-	-	-
17	-	-	-	-	-
18	-	-	-	-	-
19	-282.503	-282.094	-307.898	-322.706	-323.400
20	-	-	-	-	-
21	-	-	102.631	116.506	125.148
22	49.342	49.993	-	-	-
23	-	-	55.479	68.253	69.140
24	-	-	-	-	-
25	12.343	12.340	8.540	7.367	7.520
26	-0.006	-0.006	-0.013	-0.015	-0.015
27	-	-	0.060	0.072	0.072
28	-94.383	-94.860	-192.765	-181.853	-179.748
29	-38.251	-37.380	-	-	-
30	-0.001	-0.001	-0.001	-0.001	-0.001
31	-	-	-	-	-
32	-	-	95.821	60.290	53.804

Table 19 continued. Coefficients for Variables in 10 Best Subsets Models

VARIABLE X ( )	COEFFICIENTS MODEL - 6	MODEL - 7	MODEL - 8	MODEL - 9	MODEL - 10
1	-	-	-	-	14.852
2	48.314	47.990	48.116	42.225	90.544
3	43.180	42.196	43.004	36.314	71.472
4	-29.091	-23.736	-28.638	-31.943	-20.492
5	-	0.001	0.001	-	-
6	-19.687	-	-19.722	-12.471	-20.001
7	7.471	-	7.422	3.154	7.366
8	1.214	0.742	1.219	0.777	1.229
9	-43.276	-47.531	-43.324	-50.688	-45.938
10	28.259	26.314	27.950	16.330	26.297
11	77.730	69.766	77.653	73.638	76.891
12	67.464	68.523	67.399	56.557	65.042
13	0.579	0.608	0.574	0.632	0.551
14	48.416	38.632	49.463	45.075	48.088
15	0.000	0.000	0.000	0.000	0.000
16	0.000	0.000	0.000	0.000	0.000
17	-	-0.006	-	-	-
18	-	-	-	-	-0.038
19	-456.723	-478.352	-454.700	-464.700	-525.972
20	96.453	0.0	94.036	87.769	92.379
21	106.012	99.745	106.891	120.565	109.693
22	180.294	86.812	180.208	74.569	178.397
23	-	-	-	43.417	-
24	-8.343	-	-8.241	-5.521	-8.332
25	10.540	13.336	10.297	9.091	10.704
26	-0.007	-0.012	-0.007	-0.013	-0.007
27	-	0.044	-	0.053	-
28	-	-76.395	-	-154.000	-
29	-6.275	-10.657	-6.503	-	-5.167
30	-0.001	-0.001	-0.001	-0.001	-0.001
31	-144.244	-83.409	-144.095	-	-142.038
32	-	3.348	-	-	-

Since the units of measurement for the quantitative variables differ greatly, the magnitude of any given coefficient should not be construed solely as an indicator of influence. The units of measurement of any coefficient are the units of the response variable divided by the units of the regressor variable. The coefficients are determined jointly and serve to normalize variable values as well as measure the impact of individual variables on the response. That is, a coefficient indicates the influence of a given factor when that factor is considered simultaneously with all of the other factors in the model. A regression coefficient measures the expected change in performance per unit change in the regressor, given that the levels of the other regressors in the model remain constant. This can obscure our understanding of the role of individual factors in the model if inferences are made about these factors in isolation. That some or many of the relationships may not transfer from the composite model to the single factor situation should be understood. Since the models described here comprise on the average 20 variables, it is of particular importance that this caution be observed. A minimum of 15 variables are required to explain performance when all the control factors are being manipulated.

Selection of Explanatory and Prediction Models

Table 20 presents a rank evaluation of the ten best subset models based on explanatory R^2 and prediction R^2. With regard to quality of fit, all ten models are equivalent. It is apparent that the best explanatory models and the best prediction models do not coincide. The prediction R^2 is in all instances lower than the explanatory R^2. This

is to be expected. An *explanatory model* is one which provides an adequate fit to the data on which it was built, and in which the regression coefficients are reasonable estimates of the effects of the predictor variables. A model that is a good predictor is generalizable; that is, it provides reasonable predictions of fresh data not used in the parameter estimation process.

The models developed in this research are all linear. They serve a factor screening function and as such perform very well. Obviously higher R^2 values could be obtained if polynomial or other nonlinear models were fit. Such increases in model complexity are warranted only when they are grounded in physical reasons outside the data. This is certainly not the case here, as there is little, if any, underlying theory connecting the factors studied in this research to the performance response variables. Furthermore, in an experiment with only two levels of most factors such as this one, polynomial or nonlinear models are not meaningful.

The degree to which a model is satisfactory as a descriptor makes no implication concerning its generality. When a fitted model is applied to new data, it is unlikely to predict the fresh data as well as it fits the estimation data. Since the model is fit to the estimation set using least squares, it is, in some sense, an optimal fit for that data. Optimality here is unique to the estimation data set. Generally, a model which is 80 to 90 percent as satisfactory in prediction as it is in explanation is considered "acceptable" (20). Model 7 is only 6.4% as good in prediction as it is in description. Model 8, on the other hand, is 82.6% as good a predictor as it is at

explaining the data on which it was built.

The determining factors for model selection are model adequacy, generality and ease of use. Before model selection, then, the latter two criteria should be considered. Since those models which rank highest on explanatory R^2 are not the same as those that rank highest on the predictive scale, we know that the "best" models are not the most general. A decision must in this case be made to either 1.) use two separate models for description and prediction or 2.) use a single model which compromises between these applications. If two models are to be used, the most likely choices would be the models which rank highest on the two adequacy scales. These would be Model 7 for description and Model 6 for prediction. If a single model is to be chosen the most likely candidates are Models 10 and 8. The difference between Models 10 and 8 on the explanatory scale is +.004. The difference between Models 10 and 8 on the prediction scale is -.05272. This difference in predictive capability suggests that Model 8 would be preferable to Model 10 as a general model. In fact, Model 8, as stated above, is 82.6 percent as good in prediction as description.

The remaining major consideration for model selection is ease of use. Our focus here will be limited to the four highest ranking models on the two R^2 scales. The consecutive numbering of Models 1 through 10 corresponds to their ordering with regard to number of regressors. Model 1 has 15 regressors while Model 10 has 26. Since the top four models in terms of explanatory or prediction R^2 have at least 24 regressors, model size is not a determining factor toward ease of use.

To aid in choosing between two models or a single model, i.e. 7

and 6; or 8 or 10, we refer to Table 20 to compare the variables that comprise each model. From this table we see that the only variable *which is in Model 8 which is not in Models 6 or 7 is* X_6. *The only* variables which are in Model 10 which are not in Models 6 or 7 are X_1 and X_{18}. X_1 is an indicator variable, therefore, its presence does not affect ease of use. X_6 refers to node communication capacity. It is a direct measure which is trivially obtained. X_{18}, available communications capacity after initial assignment, is indirect and consequently more difficult to measure or estimate. This variable, which is present in Model 10, is the only one which differentiates the choices of Models 6 and 7; or Model 8 or 10 on the basis of ease of use.

If a choice is to be made between Models 8 and 10, Model 8 would be chosen on the basis of adequacy, generality and ease of use. Model 6 is 82.6% percent as good a predictor as Model 7 is at explanation which is exactly the same generality rating as Model 8. The difference between Models 6 and 8 on explanatory R^2 is 0.0026 and between Models 7 and 8 on prediction R^2 is 0.0028. Thus, it appears that Models 6 and 7 or Model 8 are essentially equivalent with respect to all evaluation criteria. Since it is usually considered preferable to use one model rather than two when all other attributes are constant, Model 8 is selected for use as the most satisfactory model for operational survivability and performance.

Table 20. Rank Ordering of 10 Best Subset Models

MODEL NO.	EXPLANATORY R^2	MODEL NO.	PREDICTION R^2 TRIM
7	.8652	6	.71536
9	.8647	8	.71252
10	.8641	2	.70286
8	.8626	10	.66264
6	.8622	1	.53577
4	.8559	9	.50232
5	.8551	3	.47850
3	.8457	4	.33184
2	.8326	5	.33173
1	.8317	7	.05569

Discussion of Model Components

Before discussing in specific the inference of model components, two *unique* aspects of this research should preface. First, it should be noted that the experiment is conducted on highly stressed distributed systems. That is, the conditions imposed were exaggerated in order to test multiple aspects of influence. These severe conditions on processing resources, application system demands, etc. were such that little modifications would force the system to failure. Some treatment combinations leave the distributed system so highly packed that after loss of a small percent of the network resources it is extremely difficult, no matter what distribution policy is imposed, to recover. While highly stressing the distributed system allows us to *determine the importance* of certain factors, it sometimes requires special understanding of model components.

The second preliminary remark pertains to definition of the regressor variables. As was indicated earlier, it is hard to measure many of the attributes of distributed systems which are used in this research. However, in light of this difficulty and the large amount of controversy which surrounds measurement of software, performance, and distributed systems, the models developed here show profound stability (24). The variables as described serve the model building process very well and as will be shown function in a very comprehensive fashion.

Now let us examine the role of the quantitative variables in the 10 best subset models. X_4, number of nodes in the distributed system, which is in all models, has a negative coefficient. This is interpreted to mean that the more nodes there are in the distributed

system the more likely that performance will be satisfactory. As can be seen in Table 19, inverse relationships of this type exist between a number of the regressor variables and the response. These instances are discussed below.

Also examination shows that some of the regressor variables have positive coefficients which indicate a direct relationship with the magnitude of the response variable. Remembering that an increase in *the value of the response means performance is moving toward failure*, we interpret strong positive relationships as having a detrimental effect on performance and consequently on operational survivability. X_7, number of application system modules, suggests that performance will degenerate as the number of application system modules increases. X_{14} simply states that failure to initially assign the application system to the distributed system makes satisfactory performance difficult.

X_{20} represents the ratio of memory requirements to useable memory capacity. The positive coefficient here says that as the memory requirements approach the total available memory, the likelihood of satisfactory performance decreases. A similar observation is made for X_{21} which represents the ratio of communications requirements to useable communications capacity. X_{22} designates application system connectivity. Its relationship to the response states that the higher the level of application system connectivity the poorer the prospects for satisfactory performance. Each of these relationships seem *reasonable and confirm some of our intuitions about distributed systems.*

X_6 indicates that the greater the capability of nodes to communicate with other nodes the more likely performance will be satisfactory. Since in this experiment the capacity of all links are held constant the communication capacity is determined strictly as a function of number of links.

Given this relationship between survivability and number of links, one would also expect distributed system connectivity to be an influential factor. Distributed system connectivity is represented by X_{19}. As expected, this factor is found in all models and in all cases *demonstrates a strong inverse relationship to response.*

To demonstrate how potentially misleading it is to interpret individual regression coefficients in a multiple regression framework, let us consider the case of X_1, X_2 and X_3 which together represent the four distributed system topologies. These topologies are star, ring, network, and array and are represented by indicator variables X_1, X_2, X_3 as discussed in Chapter IV. The coefficients for models four and five indicate that a change from the base topology, a star, to the ring topology will have an improving affect on performance. Models six through 10 further indicate that a change from the star to either the network or array would have a detrimental effect on performance. Figure 7, however, shows that average performance actually improves, although perhaps slightly, by a change from the star to any other topology. The model coefficients indicate the effect of topology given all the other factors in the model. For example, given that distributed system connectivity is represented by two quantitative variables, X_6 and X_{19}, as discussed above, it might appear less

striking that the qualitative variable, distributed system topology, X4 which also represents this same feature, has a less profound inference than expected.

DISTRIBUTION POLICY \ DISTRIBUTED SYSTEM TOPOLOGY	STAR	RING	NETWORK	ARRAY	AVERAGE
RANDOM	3.29	1.75	1.77	3.31	2.53
UNIFORM	3.00	3.08	3.25	3.00	3.08
PACKED	3.74	3.76	3.95	2.89	3.59
OPTIMAL SPARE	3.81	4.00	3.25	4.00	3.77
AVERAGE	3.46	3.15	3.06	3.30	

Figure 7. Average Performance Given for Different Distributed System Topologies and Distribution Policies

When examining the effect of distribution policy, it is observed that distribution policy in all cases is important. A change from the random distribution to any other distribution effects a noticeable positive influence on the coefficient for the factor representing the new distribution approach. While Figure 7 bears this out, it also shows that with only two exceptions performance based on distribution policy is fairly uniform. And, performance based on the intersection of distributed system topology and distribution policy is even more homogeneous. That these factors are important and that on direct observation they seem indistinguishable appear contradictory. However, once again, it must be recognized that the importance of these factors comes from their role in the model when operating with numerous other factors.

X_{30}, X_{31}, and X_{32} represent interactions between other regressor variables. X_{30} signifies the interaction among a number of application system related attributes, namely; number of application system modules, module processing requirements, module memory requirements and module communication requirements. X_{31} and X_{32} signify the interaction between final dispersion, application system connectivity and distributed system connectivity. The sign attached to interaction variables is not as important as relative magnitude of the coefficients and the signs and magnitudes of the main effects of the variables in the interaction. That these features are important to performance seems reasonable and supports the initial postulate of this research concerning the believed complexity of adequate models of operational survivability.

Several factors included in some of the 10 best subset models have negligible effects. Interestingly, these factors X_5, X_{15}, X_{16}, X_{17}, X_{18}, X_{27} (for definitions see Table 16) all represent direct measures such as node processing speed and global resource capacity, i.e., memory, processing, communication. In absolute or simple form these measures are not very meaningful, however, as has been shown, X_{24} and X_{30}, and as is shown, X_{20}, X_{21}, and X_{25}, these direct measures when considered in conjunction with other attributes of the system can be extremely influential.

Another aspect of model inference which merits a word of caution is interpretation of the signs of regression coefficients. Frequently the signs of regression coefficients will coincide with prior expectation. Most often this occurs when 1.) all necessary regressor variables are in the model, 2.) the relationship between the variables and the response is strong and 3.) the regressors are orthogonal. Models with supersets and subsets of these constraints often demonstrate this status with coefficient signs which are counter intuitive. Such inconsistencies usually are minor if they pertain to the less important factors in the model.

One of the most common causes of "wrong" signs is multicollinearity. Multicollinearity refers to the existence of intercorrelation between the regressor variables. The eleven factors involved in the 2^{K-P}_{V} Fractional Factorial experiment used in this research are orthogonal. However, a number of other candidate regressor variables were analyzed during the model building process. Many of these variables were derived from the original factors and

represent that factor in a somewhat specialized context. When all regressor variables used in the ten fitted models are analyzed simultaneously moderately strong multicollinearity is indicated. The most obvious solution is, of course, to simply remove the regressors involved in the multicollinearity. Removal of these variables, however, would destroy the predictive character of the model. Wrong signs in regression problems often occur for other reasons. For example, the violating factor may not be varied over a sufficiently wide range or necessary companion variables may be missing. The latter condition happens because a regression coefficient is a measurement of partial effect and does not stand alone. This type of wrong sign condition can sometimes be "corrected" by a redefinition of the variable.

It is not necessary that the signs of coefficients be in agreement with prior expectation. The degree to which the factors fall short of fulfilling their combined role of explaining or predicting the response is reflected in the model adequacy evaluations. Adjustments which influence the direction of signs such that they concur with expectation may result in models with higher adequacy ratings. This does not imply, however, that sign concurrence will assure an increase in model adequacy or that a model that fits a set of estimation data will have intuitive appeal or be a useful predictor of new observations. For further discussion see Mentgomery and Peck (20).

Inferences of some model components require thoughtful interpretation. Factors X_{26} and X_{28}, for example, are to some extent related, however, not enough to be determined redundant. Redundancy

would indicate that one of the variables could be removed without affecting the model. Here, we find that either both X_{26}, available processing capacity at the end of the subcase, and X_{28}, dispersion at the end of the subcase, are present in the model or just X_{26} is. X_{28} represents the final number of nodes over which the application system is distributed divided by the number of application system modules. The closer dispersion comes to being total, or one, the more likely performance is to be satisfactory. The potential for dispersion is, of course, related to the resources available. Thus, a somewhat collinear relationship between X_{26} and X_{28} is to be expected.

For purposes of inference an interrelationship exists between all the regressor variables that relate either directly or indirectly to dispersion.

X_{23} represents dispersion after initial assignment. It is implied that the greater initial dispersion that exists the less likely performance will be satisfactory. The apparent contradiction between the relationship of X_{23} to the response and that of X_{28} to the response requires further investigation. The coefficient on X_{23} infers initial dispersion is "bad" and the coefficient on X_{28} infers final dispersion is "good." The phenomemon observed here results from the highly stressed nature of this experiment. That is, the design of the distributed systems tested was such that if a small number of nodes were lost, features of the system other than simply excess processing and memory capacity were required to make it survive. Thus, if the application system was initially dispersed there would be an increased probability that losing nodes would make it impossible to recover. If,

on the other hand, the application system was concentrated on a few nodes, the likelihood of losing valuable nodes would decrease resulting in a higher probability of survival.

This situation is further exhibited by X_{29}, criticality of the lost nodes. Criticality here is determined as the ratio of the sum of the connectivity of application modules on the lost nodes to application system connectivity. The coefficient states that the closer the criticality ratio comes to one the more satisfactory performance. Although this relationship is worthy of further study, the following is offered as a possible explanation. Previously it was postulated that final dispersion has a constructive relationship to performance. This factor, however, seems to imply that when examining lost nodes concentration is desirable. It is possible that both of these conditions hold, however, one pertains to performance and the other to likelihood of satisfactory reconfiguration. The models infer that initial concentration of the application system is desirable. It follows, then, that concentration of that which is lost will facilitate recovery.

X_{24} represents memory consistency, that is the number of application system modules divided by the number of application system modules that will "fit" on a node memory-wise. This says that as this ratio increases chances for survival improve. In other words, the closer the system can come to placing all the application system modules on a single node, the less likely that performance will be satisfactory. Once again, if we relate dispersion to performance and concentration to recovery, this inference is reasonable. The fewer the

application modules that will fit on a node the more likely that the application system will be dispersed.

X_{25}, processor consistency, represents the ratio of number of application system modules to average number of modules which will "fit" on a node processor-wise. The positive coefficient here indicates that as this ratio increases performance degrades. Such a proposal is intuitive. The probability of satisfactory application system performance will increase directly with the ability to assign all processing to a single node. However, when capability falls short of that, the importance of distribution policy and connectivity may become dominant. It may not be that high processor consistency is detrimental when the ratio is greater than one but that in complex systems reconfiguration is difficult.

It is apparent that dispersion and concentration are companion concepts. Further analysis using some of these factors in a designed experiment so that their main effects and interactions may be more precisely estimated is desirable. Exercising these factors in less highly stressed experiments may be necessary. Experiments of this type should be useful in clarifying the relationship between these variables and the response.

The research documented in this dissertation demonstrates both the capability and significance of empirical investigation in distributed processing. The experimental results presented do not support conclusions drawn from prior analytical models (19). Merwin and Mirhakak's survivability index, for example, indicates that the most survivable distributed network is a star. That determination is

based on number of links to be traversed between any two network nodes. This research clearly shows that other factors such as potential for alternate routes, characteristics of the application system and distribution method are also strongly influential. When averaging responses based on topology and distribution policy all three other topologies tested faired better than the star topology. Differences are further highlighted when all model components are considered. These results bring into question the present capability of analytical models to represent complex problems of inexact sciences. It is also apparent, however, that analytical modeling may be appropriate for examination of specific model components such as those which can be expressed in totally quantitative terms. The three consistency measures fall into that category. Empirical methods with which to test analytical models are available. Used together, these methods should lead to a strongly quantitative understanding of survivability which can be used in the design of distributed systems and validated in field tests.

CHAPTER VIII

CONCLUSIONS AND RECOMMENDATIONS

One objective of this research was to enhance our understanding of operational survivability and performance and to make that understanding quantitative. The approach taken was an experimental one which used factor screening to give indication of variable importance. The objective was to develop models which are explanatory and would provide a foundation for future refinements rather than perscriptive. The second objective of this research was to demonstrate the applicability of traditional experimental design and regression analysis techniques to the field of computer science. The experiment and results documented in this dissertation support these objectives.

A factor screening experiment was conducted to determine whether any of a large set of candidate regressor variables were important to operational survivability and performance. Results demonstrate that a number of variables are, indeed, very influential and analysis shows their approximate level of importance. A two level factor screening design with a large number of variables was used in this research. Given this experimental approach, it is relatively unusual and encouraging that the design provides sufficient information on which to build ten linear explanatory models with R^2 values in the range of .8. Further, the capability of several of these models to serve exceptionally well in a predictive role suggests that they provide a good foundation on which to build future refinements.

The models developed here through standard regression techniques make a number of statements about measurement of operational survivability and performance. The first and most important statement is that these attributes can, in fact, be described in a quantitative fashion. Next, they imply that certain factors are more important than others in determining the level of the response. Some of the most influential factors are distributed system connectivity, number of nodes, available processing capacity, distribution policy, application system connectivity and module memory requirements. The number of regressor variables required to achieve explanatory model adequacy levels of .8 is large. Large is here defined as between 15 and 26. The nine core variables found in all ten models have the expected sign and an obvious interpretation. The few instances in which signs do not concur with expectation occur in connection with peripheral or less important factors. These instances are well within acceptable bounds for research of the type conducted here. It is further shown that among the nine essential factors are factors which represent the three general categories *hypothesized at the outset* of this research. Also, it is demonstrated that no single category or pair of categories will adequately explain or predict operational survivability or performance. The three categories describe attributes of the distributed system network, application system and distribution policy.

Analysis of the experiment results supports the hypothesis that the factors necessary to adequately describe operational survivability would be large in number and non-trivial in observation. The ten best subset models included a number of factors which were nominal

in influence. Each of these are directly measureable entities such as global memory, processing and communications capacity. The more important factors tended to be more complex or more indirectly derived. *Examples are* distributed system connectivity, application system connectivity and memory consistency. This finding further supports the initial proposal that operational survivability cannot be trivially indexed.

In summary, 32 candidate regressors are used in identifying the 10 best subset models. The *coefficients of these regressors are* approximately equivalent in sign and magnitude across models. All variables remain proportional with the introduction and removal of other variables, thereby demonstrating extreme stability. The explanatory adequacy of models built using these variables is in all instances in excess of .8 which is very acceptable for a factor screening experiment. The adequacy in prediction of these models ranges *between* -.39 *and* +.71 with some models predicting very well and others predicting very poorly. By constructing satisfactory explanatory and predictive models, this research demonstrates that the concept of operational survivability and performance as proposed can be expressed quantitatively. Further, it is shown that major factors include the distributed system network, application system and distribution policy as initially proposed.

In review we find that there are nine factors found in all models. These are number of nodes in the distributed system, distributed system connectivity, module memory requirements, module to module interaction frequency, distribution policy, percent nodes lost,

initial assignment results, available processing capacity at the end of the subcase and the interaction of all application related variables.

Other factors which prove to be important and function in the models in an expected manner are number of application modules, node communication capacity, memory requirements ratio, communication requirements ratio, and application system connectivity. Some factors operating as expected given the highly stressed nature of the experiment conducted are initial and final dispersion; memory and processor consistency; and criticality of lost nodes.

Factors having negligible effect include node processing speed; global memory capacity; available processing, memory and communications capacity after initial assignment; and available communications capacity at the end of the subcase.

A number of propositions can be inferred from the analyses of Chapter VII. Some of these are not unexpected. Others, however, are somewhat surprising, and we offer them as hypotheses which can be further explained experimentally. While plausible explanation can be offered to support each of these hypotheses, there are also apparently plausible settings in which the hypotheses may fail. Both confirming instances and refutations of the hypotheses point the way toward further experimentation. That is, for each of the 10 hypotheses listed below we present a possible mechanism to explain the effect which is apparently being observed. We then give a brief indication of situations in which the hypothesis may fail; the appropriate experimental setting for dealing with the hypothesis should lie within these limits.

1.) The more nodes there are in the distributed network, the more likely that performance is satisfactory. It seems very likely that the distributed systems in which we are most interested satisfy this property, that is the more nodes there are in a distributed network configuration the more likely there will be slack or excess resource capacity which can be used if other resources are lost. However, given a ring network configuration with communication links traveling in only one direction, the loss of a single node will destroy the network no matter how many nodes it contains. Likewise, this is true for a star configuration if the central node is the node lost.

2.) As the memory requirements approach the total available memory, the likelihood of satisfactory performance decreases. Given an application system which is distributed over the nodes of a distributed network, it is reasonable to conclude that as the demands on memory approach the memory limit of the network the more likely additional resource losses will have a detrimental effect on survivability due to constraints on reconfiguration options. On the other hand, it is apparent that as the distributed network decreases so too does the available memory until finally the memory available is only that on a single node. Further, it is possible that the memory requirements of the application system are extremely low and fit well within the memory capacity of a single node, however, the processing demands

exceed the capabilities of the processor. In this case the memory requirements to availability ratio has no relationship to survivability.

3.) As the module interaction or communications requirements approach the total available communications capacity, the likelihood of satisfactory performance decreases. It is not difficult to envision a number of network configurations in which the options for satisfactory reassignment of application modules decreases as the communications demands of the application system approach the communication limit of the distributed system. However, if the interaction requirements of application modules is such that those modules having the highest interaction can always be placed on a single node this relationship may not hold. Also, if the network configuration is such that two large subnetworks are connected by a bridge and the application system is split such that a large portion of the module to module interactions must traverse the bridge, the performance may decrease even though the available communication capacity is high.

4.) The higher the distributed network connectivity, the greater its probability of survival. Research in network survivability and routing support our basic intuition that *in general* the larger the *number* of alternate routes available for nodes to communicate with other nodes the greater the likelihood that an application system spread

over several nodes will be able to continue to adapt to increased node losses. *Special cases can be identified* to which this general statement does not apply. For example, if a network comprising nodes and links of low capacity or nodes and links which are nearly saturated is highly connected and the distribution/redistribution policy is such that tasks are *dynamically reassigned* to "*optimize*" *node and* link utilization, the fact that the options are numerous may *be a drawback. In other* instances high network connectivity may be irrevelent to survivability. For example, if a network is highly connected but the application system to be executed on *it* comprises *only two modules*, *the degree* of network connectivity may be of negligible importance.

5.) Failure to properly assign the application system to the distributed network initially makes satisfactory or degraded performance difficult. The complexity of mapping an application topology onto a network topology increases with the size and connectivity of the two graphs to be mapped. Thus, when an application system is assigned to the distributed system in such a way that it does not meet performance requirements, adjustments to correct the problem require additional sophistication on the part of the redistribution strategy. That is, once the problem of unsatisfactory performance is detected, the cause must be determined and a solution found. Depending on the distribution/redistribution policy the solution space for correction is often

more constrained than the initial solution space. On the other hand, if the distribution/redistribution algorithm is an adaptive one that examines different distribution options to determine their effect, then unsatisfactory initial allocations may be more useful or informative than satisfactory distributions. Unsatisfactory distributions may provide insight into worst case conditions.

6.) Performance degenerates as the number of application modules increases. We are essentially postulating that as the number of application modules increases the task of assigning and reassigning them in such a way that performance is satisfactory becomes increasingly more complicated. This is particularly true if the modules have high interaction requirements and few options for assignment due to module size or network configuration constraints. To see how such a mechanism could fail to hold, let the application system be of size N. The choice exists to either have five modules of size N/5 or 20 modules of size N/20 on a 10 node network, any node of which can accommodate an N/4 size module. It is clear that having more modules offers more flexibility and possibly more opportunity for satisfactory assignment. Here, the larger number of small modules potentially fit on four nodes and could disperse to 10 nodes. The larger modules need at minimum five nodes. Maximum dispersion for the larger modules is also five nodes.

7.) The higher the level of application system connectivity the poorer the prospects for satisfactory performance. Here again, there is indication that high software system complexiity will influence survivability. Given an application system for which module requirements nearly correspond to individual node capabilities, high connectivity may require a one for one mapping of the application system onto the distributed network. If such a mapping can be constructed initially it is unlikely that it can be maintained with increasing node losses. There are also instances in which software complexity may have little or no effect on survivability and performance. For example, an application system can be highly connected but have module to module interaction frequencies so low that as long as there is a path from any module to any other module the interaction demands can be met.

8.) The greater initial dispersion the less likely performance will be satisfactory. Given a distributed network of high or low connectivity it is not difficult to find situations in which the greater initial dispersion the more difficult recovery due to reduced reconfiguration options. Depending on the distribution/redistribution approach, however, it may be that the greater initial dispersion the fewer application modules to be reassigned after the loss of any single node. This would indicate that initial dispersion has a positive influence on performance.

9.) The greater final dispersion the more likely performance will be satisfactory. Corresponding to the previous inference, the greater the initial flexibility in the system the greater the opportunities for subsequent reconfiguration. The greater final dispersion the less saturated the system resources. While this argument may hold it is also possible to imagine application systems for which the postulate may not be true. For example, the greater final dispersion the less likely that highly connected application systems with high average module to module interaction frequency will be assigned such that their performance requirements can be met.

10.) The larger the proportion of highly connected application modules on the nodes lost the greater the likelihood of survival. Like hypothesis (8), this hypothesis concerns the effects of possible reconfiguration options. The effect in this case is simply one of removing logical dependencies: as dependencies are removed, the remaining nodes become (if they still meet the application requirements) autonomous and this can be exploited in assigning the remaining resources. Again, special cases can be described for which this argument does not hold. One such case is that in which the modules to be reassigned are highly connected but the network onto which they are to be placed is heavily saturated and available resources are widely dispersed. Given these conditions it is likely that performance will

degrade rather than improve. Thus, it appears having the flexibility to reassign all modules having the most severe interaction constraints can facilitate successful reassignment given the network resources are not heavily saturated.

The experimental and modeling techniques used in this dissertation represent an initial step in developing a measurement instrument for operational survivability in gracefully degrading distributed processing systems. Further refinements in this measurement tool should facilitate more precision in its explanatory and predictive capability. Other recommendations for future research fall into two categories. These categories are 1.) more extensive use of the simulator as an experimental device and additions to its current capabilities and 2.) experimentation to clarify the operation of specific factors. SURSIM is a fairly general purpose simulator. The parameter levels used for this research designate selections made for this factor screening experiment. They do not represent limitations of the simulator. Using the variables under its control the simulator can generate a virtually unlimited number of treatment combinations. This provides the capability to focus future experimentation on some single or small set of factors while fixing the context environment with appropriate constants. There are features of the simulator which were not exercised in this experiment. One of these was to vary the capacities of the communication links. Also, heterogeneous distributed systems can be described and accommodated by the simulator manipulation and evaluation routines. Among the possible additions to the simulator

are the capability to represent multiple communications links between nodes; limitation on the availability of software; node and link vulnerability and criticality; and a larger number of distribution policies.

Future research which is likely to be productive includes experimentation on factors related to dispersion, connectivity and distribution policy. Designed experiments which focus on the control of these factors should improve our understanding of their direct and *indirect operation. The introduction of more interaction variables* could also be helpful. Also, exercising the factors examined in less highly stressed experiments may be meaningful.

The research conducted here identifies the variables important to *operational survivability and* to some extent tells how large changes in these important variables affect the response. Future experimentation which provides either a large number of factor levels or finer granularity in possible variable values should permit greater resolution in the simulator results *and their subsequent application.* The results presented in this dissertation demonstrate the applicability of traditional experimentation and regression analysis in the field of computer science as well as the feasibility of measurements which can serve as measurements for *distributed systems. The models* developed represent a promising initial step in the quantification of operational survivability as it applies to gracefully degrading distributed processing systems.

APPENDICES

APPENDIX A

DESCRIPTION OF DATA USED IN DESIGNED EXPERIMENTS

A description of data used in the 128 designed experiments is presented below.

1. Factor Z_1 - Distributed System Topology:

   Four different topologies are used in this experiment. They are a star, ring, network and array. Examples of these topologies for four and 10 node networks are presented in Figures A-1 through A-4.

2. Factor Z_2 - Number of Nodes:

   Two different size distributed networks are used in this experiment. These comprise 4 and 10 nodes respectively.

3. Factor Z_3 - Node Processing Speed:

   Two node processing speeds are used in this experiment. They are 500 kilo operations per second or 500 kps and 10 million operations per second or 10 mops.

4. Factor Z_4 - Node Memory Capacity:

   Two different node memory capacities are used. These are 128 kilobytes or 128 kbytes and 2 megabytes or 2 mbytes.

5. Factor Z_5 - Connectivity of Applications System:

   Application systems with high and low connectivity are used. The topology of these systems for the 4 and 16 node application systems used in this experiment are presented in Figures A-5 through A-8.

6. Factor Z_6 - Number of Application Modules:

   Two different quantities of application system modules are used in this experiment. They are 4 and 16 modules respectively.

7. Factor Z_7 - Average Module Processing Requirements:

   Application module processing requirements are computed as .1 or .5 of the node processing capacity after total network processing capacity is divided by the quantity of application system modules.

8. Factor Z_8 - Average Module Memory Requirements:

   Application module memory requirements are computed as .1 or .8 of the node memory capacity after total network memory capacity is divided by the quantity of application system

modules.

9. Factor Z_9 - Average Module to Module Interaction Frequency:

Two levels of interaction frequency are used. These are high interaction frequency, which is computed as 50% of the average module processing requirements; and low interaction frequency, which is computed as 1% of the average module processing requirements. These frequencies are expressed in thousands of messages or packets sent per execution of an application module.

10. Factor Z_{10} - Distribution/Redistribution Policy:

Application system modules are assigned to the distributed system topologies according to one of four possible graph mapping algorithms. The algorithms used in this experiment are defined as follows.

Random Distribution - Application system modules are randomly assigned to processors. If the application module and communication burden will not fit at the node selected another random assignment will not be made. This will be repeated until all modules have been assigned to node. Should this approach fail to construct a map, the simulator in its present form will not attempt to degrade or reconfigure the system.

Uniform Distribution - Application system modules are assigned to nodes such that each node has as near the same operating demands as possible. This type of distribution is relatively easy to implement in central processor or master/slave type systems. Distributed systems in which global information about the system is availabl- to each node must take into account the overhead burden this will place on the system resources. The overhead burden is dependent upon the size of the distributed system and timeliness of information required, i.e., frequency of update. (In distributed systems with high capability nodes, the impact of this update activity may be negligible. For distributed systems with a large number of low capability nodes, this burden is possibly very significant.) For the simulation under discussion such overhead burden will not be a factor; however, given some rule to be used to determine overhead burden incorporation into the model would be possible.

Packed Distribution - Application system modules are assigned to a designated processor until it reaches maximum capacity after which point modules are assigned to the next (nearest) processor, etc. If multiple processors are one communication link away the next node to be packed will be randomly chosen.

Optimal Spare Distribution - Application system modules are

assigned to the distributed processing system in such a way that each node being asigned application tasks has a spare queue indicating the sequence of backup or spare nodes which will be activated should the former fail. If insufficient nodes are available to provide every node with a spare, spares will be given to the nodes with application modules having the highest criticality ranking. Other "spares" may be shared by nodes executing lower criticality software. The concept of optimal-spare will become more complex and perhaps yet more meaningful when the vulnerability attribute is incorporated into the model.

11. Factor Z_{11} - Percent Nodes Eliminated:

Four different ranges of percent node elimination are used. These are

1 - 10%
11 - 30%
31 - 50%
51 - 80%

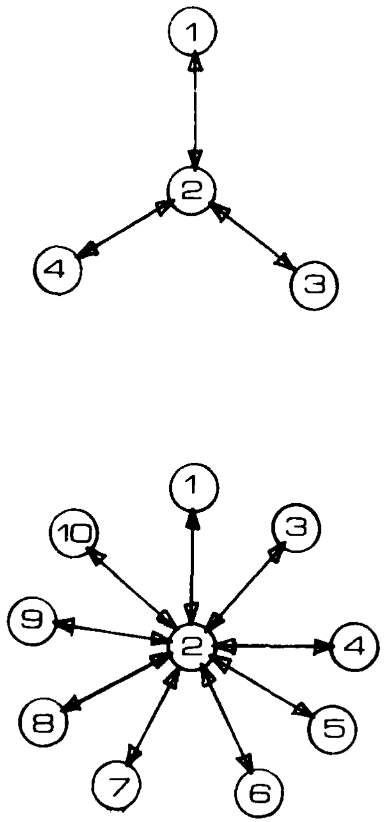


Figure A-1. Four and 10 Node Star Topologies

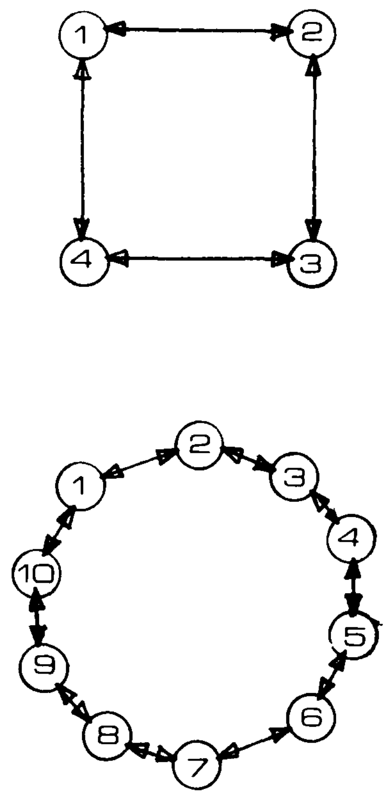


Figure A-2. Four and 10 Node Ring Topologies

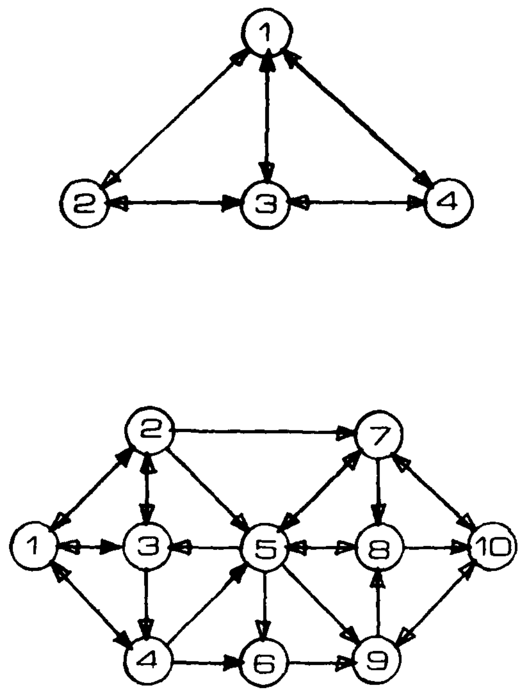


Figure A-3. Four and 10 Node Network Topologies

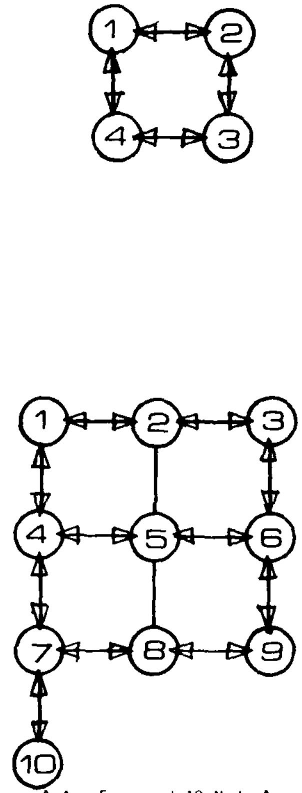


Figure A-4. Four and 10 Node Array Topologies

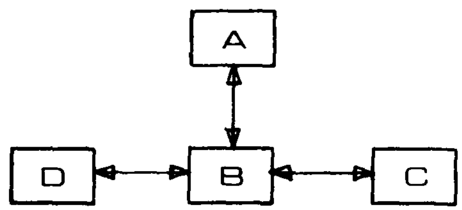


Figure A-5. Four Module Application System - Low Connectivity

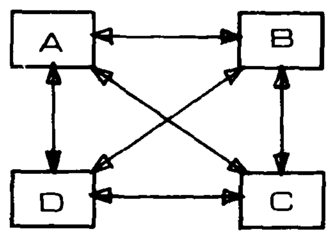


Figure A-6. Four Module Application System - High Connectivity

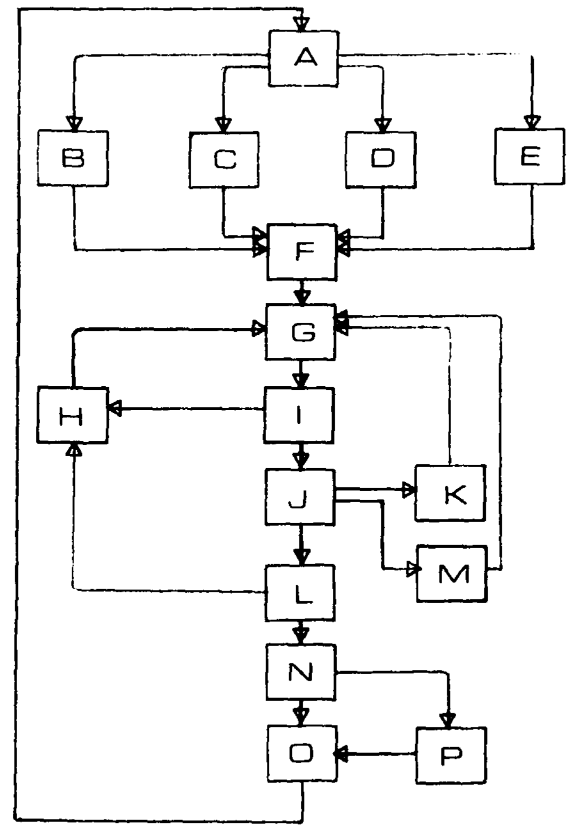


Figure A-7. Sixteen Module Application System Low Connectivity

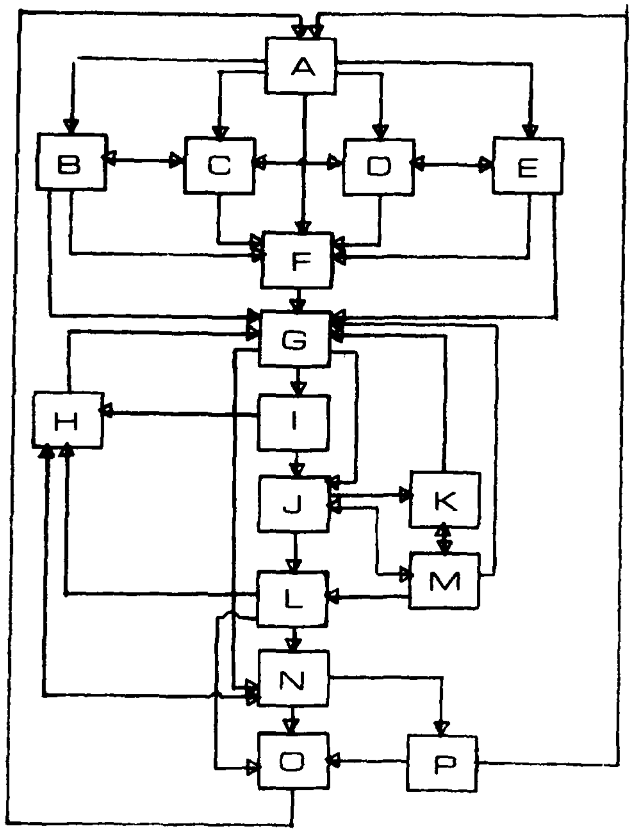


Figure A-8. Sixteen Module Application System High Connectivity

APPENDIX B

TEN OPTIMAL SUBSET MODELS

Variable Key

Factor Label Code/Variable		Factor Description
[illegible]	[illegible]	[illegible] Indicating [illegible] Topology
[illegible]	[illegible]	Number of Nodes in the Distributed System
[illegible]	[illegible]	Node Processing Speed
[illegible]	[illegible]	Node Communications Capacity
[illegible]	[illegible]	Number of Application System Modules
[illegible]	[illegible]	Module Memory Requirements
[illegible]	[illegible]	Module to Module Interaction Frequency
[illegible]	X10	Dummy Variables Indicating Distribution Policy
[illegible]	X11	
[illegible]	X12	
[illegible]5	X13	Percent Nodes Lost
[illegible]7	X14	Initial Assignment Result
R2	X15	Global Memory Capacity
R4	X16	Available Processing Capacity after Initial Assignment
R5	X17	Available Memory Capacity after Initial Assignment
R6	X18	Available Communications Capacity after Initial Assignment
S1	X19	Distributed System Connectivity
S2	X20	Memory Requirements/Useable Memory Capacity
S4	X21	Communications Requirements/Useable Communications Capacity
S6	X22	Application System Connectivity
S7	X23	Dispersion - Initial (Number of nodes over which an application system is distributed/ (Number of application system modules)
S8	X24	Memory Consistency (Number of application system modules)/ (Average number of application system modules that will "fit" on a node - memory wise)

Variable Key continued.

Factor Label Code/Variable		Factor Description
S9	X25	Processor Consistency (Number of application system modules)/ (Average number of application system modules that will "fit" on a node-processorwise)
A4	X26	Available Processing Capacity at End of Subcase
A5	X27	Available Communications Capacity at End of Subcase
A6	X28	Dispersion at End of Subcase (Number of nodes over which the application system is distributed)/ (number of application system modules)
A7	X29	Criticality of Lost Nodes (Sum of the connectivity of the application system modules residing on the lost nodes)/ (Application system connectivity)
X3	X30	Interaction between Number of Application System Modules and Module Processing Requirements and Module Memory Requirements and Module Communication Requirements
X5	X31	Interaction between Dispersion at End of Subcase and Application System Connectivity
X6	X32	Interaction between Dispersion at End of Subcase and Application System Connectivity and Distributed System Connectivity

R-SQUARED	ADJUSTED R-SQUARED	CP	VARIABLE	COEFFICIENT	T-STATISTIC
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

Optimal Subset Model # 1

R-SQUARED	ADJUSTED R-SQUARED	CP	VARIABLE	COEFFICIENT	T-STATISTIC
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

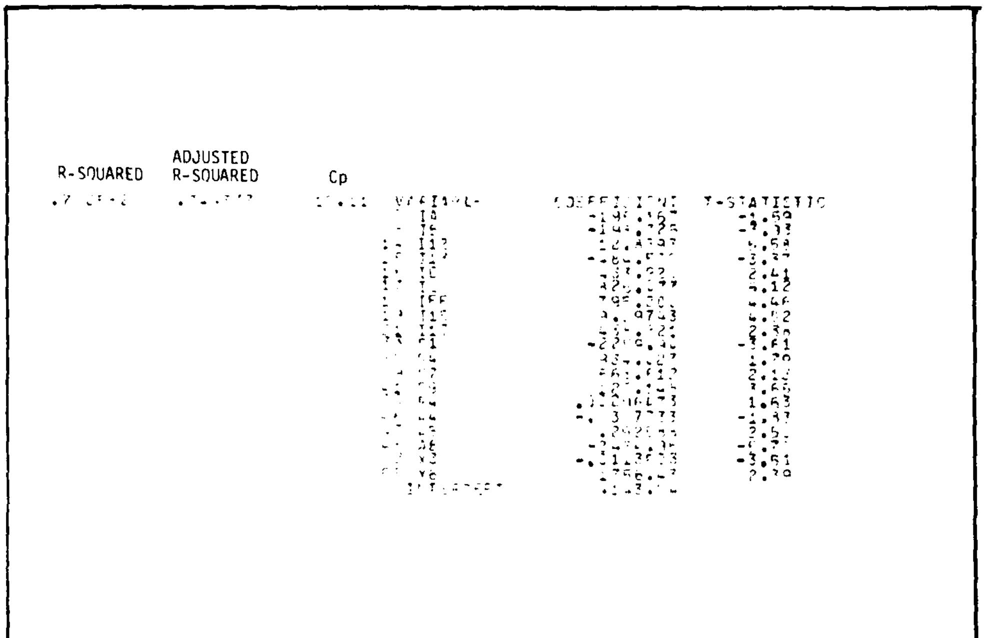
Optimal Subset Model # 2

R-SQUARED	ADJUSTED R-SQUARED	Cp	VARIABLE	COEFFICIENT	T-STATISTIC
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
			INTERCEPT	[illegible]	

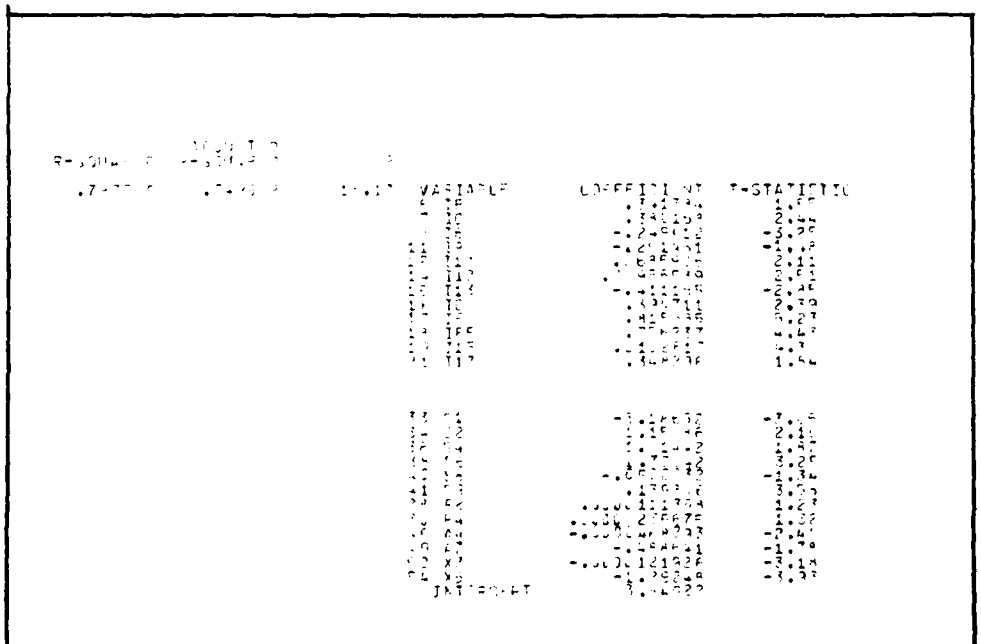
Optimal Subset Model # 3

R-SQUARED	ADJUSTED R-SQUARED	Cp	VARIABLE	COEFFICIENT	T-STATISTIC
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
			INTERCEPT	[illegible]	

Optimal Subset Model # 4



Optimal Subset Model # 5



Optimal Subset Model # 6

R-SQUARED	ADJUSTED R-SQUARED	Cp	VARIABLE	COEFFICIENT	T-STATISTIC
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

Optimal Subset Model # 7

135

R-SQUARED	ADJUSTED R-SQUARED	Cp	VARIABLE	COEFFICIENT	T-STATISTIC
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

Optimal Subset Model # 8

136

R-SQUARED	ADJUSTED R-SQUARED	Cp	VARIABLE	COEFFICIENT	T-STATISTIC
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

Optimal Subset Model # 9

137

R-SQUARED	ADJUSTED R-SQUARED	Cp	VARIABLE	COEFFICIENT	T-STATISTIC
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

Optimal Subset Model # 10

APPENDIX C

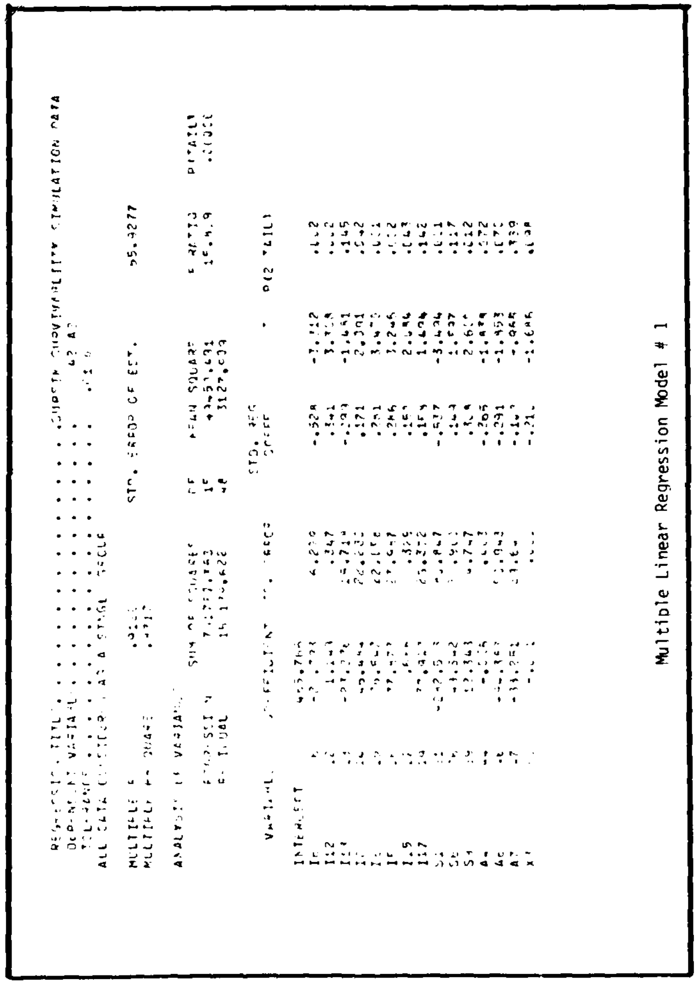
TEN MULTIPLE LINEAR REGRESSION MODELS BUILT FROM ESTIMATION SET DATA B

Variable Key

Factor Label Code	Variable	Factor Description
IA	X 1	Dummy Variables Indicating
IB	X 2	Distributed System Topology
IC	X 3	
I6	X 4	Number of Nodes in the Distributed System
I7	X 5	Node Processing Speed
I9	X 6	Node Communications Capacity
I10	X 7	Number of Application System Modules
I12	X 8	Module Memory Requirements
I13	X 9	Module to Module Interaction Frequency
ID	X10	Dummy Variables Indicating Distribution Policy
IE	X11	
IF	X12	
I15	X13	Percent Nodes Lost
I17	X14	Initial Assignment Result
R2	X15	Global Memory Capacity
R4	X16	Available Processing Capacity after Initial Assignment
R5	X17	Available Memory Capacity after Initial Assignment
R6	X18	Available Communications Capacity after Initial Assignment
S1	X19	Distributed System Connectivity
S2	X20	Memory Requirements/Useable Memory Capacity
S4	X21	Communications Requirements/Useable Communications Capacity
S6	X22	Application System Connectivity
S7	X23	Dispersion - Initial (Number of nodes over which an application system is distributed/ (Number of application system modules)
S8	X24	Memory Consistency (Number of application system modules)/ (Average number of application system modules that will "fit" on a node - memory wise)

Variable Key continued.

Factor Label Code/Variable		Factor Description
S9	X25	Processor Consistency (Number of application system modules)/ (Average number of application system modules that will "fit" on a node-processorwise)
A4	X26	Available Processing Capacity at End of Subcase
A5	X27	Available Communications Capacity at End of Subcase
A6	X28	Dispersion at End of Subcase (Number of nodes over which the application system is distributed)/ (number of application system modules)
A7	X29	Criticality of Lost Nodes (Sum of the connectivity of the application system modules residing on the lost nodes)/ (Application system connectivity)
X3	X30	Interaction between Number of Application System Modules and Module Processing Requirements and Module Memory Requirements and Module Communication Requirements
X5	X31	Interaction between Dispersion at End of Subcase and Application System Connectivity
X6	X32	Interaction between Dispersion at End of Subcase and Application System Connectivity and Distributed System Connectivity



Multiple Linear Regression Model # 1

REGRESSION TITLE. . . . . .SURSIM SURVIVABILITY SIMULATION DATA
DEPENDENT VARIABLE. 42 A2
TOLERANCE . . . . [illegible]
ALL DATA CONSIDERED AS A SINGLE GROUP

MULTIPLE R .9126 STD. ERROR OF EST. [illegible]
MULTIPLE R-SQUARE [illegible]

ANALYSIS OF VARIANCE

	SUM OF SQUARES	DF	MEAN SQUARE	F RATIO	P(TAIL)
REGRESSION	[illegible]	16	[illegible]	[illegible]	[illegible]
RESIDUAL	[illegible]	47	[illegible]		

VARIABLE		COEFFICIENT	STD. ERROR	STD. REG COEFF	T	P(2 TAIL)
INTERCEPT		[illegible]				
I6	6	[illegible]	[illegible]	-.524	[illegible]	[illegible]
I7	7	[illegible]	[illegible]	[illegible]	.516	[illegible]
I12	[illegible]	[illegible]	[illegible]	[illegible]	3.292	[illegible]
I13	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	14	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	15	[illegible]	[illegible]	[illegible]	3.466	[illegible]
[illegible]	16	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
I15	17	[illegible]	.332	[illegible]	[illegible]	[illegible]
I17	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
S1	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
S4	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
S9	[illegible]	[illegible]	[illegible]	[illegible]	2.597	.013
A4	44	[illegible]	[illegible]	-.261	-1.795	.079
A6	[illegible]	[illegible]	[illegible]	-.292	-1.947	.071
A7	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
X3	[illegible]	[illegible]	[illegible]	-.212	-1.687	[illegible]

Multiple Linear Regression Model # 2

143

REGRESSION TITLE. . . . . .SURSIM SURVIVABILITY SIMULATION DATA
DEPENDENT VARIABLE. 42 A2
TOL RANC .0100
ALL DATA CONSIDERED AS A SINGLE GROUP

MULTIPLE R [illegible] STD. ERROR OF EST. [illegible]
MULTIPLE R-SQUARE [illegible]

ANALYSIS OF VARIANCE

	SUM OF SQUARES	DF	MEAN SQUARE	F RATIO	P(TAIL)
REGRESSION	[illegible]	16	[illegible]	13.707	.00000
RESIDUAL	[illegible]	[illegible]	[illegible]		

VARIABLE		COEFFICIENT	STD. ERROR	STD. REG COEFF	T	P(2 TAIL)
INTERCEPT		[illegible]				
I6	6	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
I7	7	[illegible]	[illegible]	.134	.573	.569
I12	12	[illegible]	[illegible]	[illegible]	[illegible]	.012
I13	13	[illegible]	14.241	-.214	[illegible]	[illegible]
[illegible]	14	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	15	[illegible]	[illegible]	[illegible]	3.333	.001
[illegible]	16	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
I15	17	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
I17	19	[illegible]	[illegible]	.171	[illegible]	.141
S1	[illegible]	[illegible]	[illegible]	-.565	-3.649	[illegible]
S4	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
S7	[illegible]	55.473	[illegible]	[illegible]	[illegible]	[illegible]
S9	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
A4	44	[illegible]	[illegible]	[illegible]	-2.179	.035
A5	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
A6	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
X3	[illegible]	[illegible]	[illegible]	[illegible]	-1.518	[illegible]
X6	[illegible]	[illegible]	[illegible]	[illegible]	.861	[illegible]

Multiple Linear Regression Model # 3

144

REGRESSION TITLE. . . . . .SURSIM SURVIVABILITY SIMULATION DATA
DEPENDENT VARIABLE. 42 A2
TOLERANCE 0100
ALL DATA CONSIDERED AS A SINGLE GROUP

MULTIPLE R .[illegible] STD. ERROR OF EST. 54.0470
MULTIPLE R-SQUARE .[illegible]9

ANALYSIS OF VARIANCE

	SUM OF SQUARES	DF	MEAN SQUARE	F RATIO	P(TAIL)
REGRESSION	[illegible]	19	40179.336	13.754	[illegible]
RESIDUAL	[illegible]	44	2921.092		

VARIABLE		COEFFICIENT	STD. ERROR	STD. REG COEFF	T	P(2 TAIL)
INTERCEPT		[illegible]				
IA	3	[illegible]	16.678	[illegible]	-1.760	[illegible]
I6	6	-23.663	6.620	-.651	-3.574	.001
I7	7	[illegible]	.011	.031	.536	.594
I12	12	[illegible]	.374	.256	2.310	.026
I13	13	-57.973	18.919	-.223	-2.854	[illegible]
IC	14	12.177	26.369	.065	.462	.646
IE	15	[illegible]	21.716	.274	3.442	.001
IF	16	[illegible]	[illegible]	.198	2.128	.039
I15	17	.611	.304	.134	2.007	.051
I17	19	47.994	27.061	.193	1.768	.084
S1	21	[illegible]	92.925	-.624	-3.492	[illegible]
S4	24	[illegible]	67.033	.138	1.887	.066
S7	27	[illegible]	41.995	.183	1.625	.111
S9	29	[illegible]	4.871	.184	1.512	.138
A4	44	[illegible]	[illegible]	-.661	-2.559	.014
A5	45	[illegible]	.029	.577	2.453	.018
A6	46	[illegible]	65.187	[illegible]	-2.930	[illegible]
X3	[illegible]	[illegible]	.008	-.162	-1.438	.158
X6	[illegible]	[illegible]	[illegible]	.069	.545	.589

Multiple Linear Regression Model # 4

REGRESSION TITLE. . . . . .SURSIM SURVIVABILITY SIMULATION DATA
DEPENDENT VARIABLE. 42 A2
TOLERANCE . . . .[illegible]
ALL DATA CONSIDERED AS A SINGLE GROUP

MULTIPLE R .[illegible]7 STD. ERROR OF EST. 54.2032
MULTIPLE R-SQUARE .[illegible]

ANALYSIS OF VARIANCE

	SUM OF SQUARES	DF	MEAN SQUARE	F RATIO	P(TAIL)
REGRESSION	[illegible]	19	40139.177	13.662	[illegible]
RESIDUAL	[illegible]	44	2937.992		

VARIABLE		COEFFICIENT	STD. ERROR	STD. REG COEFF	T	P(2 TAIL)
INTERCEPT		[illegible]				
IA	3	[illegible]	16.921	[illegible]	-1.769	[illegible]
I6	6	[illegible]	6.842	-.610	-3.599	.001
I12	12	[illegible]	.376	.255	2.280	.028
I13	13	[illegible]	[illegible]	-.228	-2.836	.007
IC	14	[illegible]	26.572	.064	.449	.656
IE	15	[illegible]	21.924	.274	3.471	[illegible]
IF	16	[illegible]	[illegible]	.195	2.172	.041
I15	17	[illegible]	.306	.134	1.998	.052
I17	19	[illegible]	[illegible]	.197	1.744	.088
R4	34	[illegible]	[illegible]	[illegible]	.191	.857
S1	21	[illegible]	[illegible]	-.615	-3.877	[illegible]
S4	24	[illegible]	67.178	.196	1.963	.069
S7	27	[illegible]	[illegible]	.190	1.643	.108
S9	29	[illegible]	4.990	.184	1.507	.139
A4	44	[illegible]	[illegible]	[illegible]	-2.522	.015
A5	45	[illegible]	.030	.577	2.429	.019
A6	46	[illegible]	[illegible]	-.553	-2.749	.009
X3	[illegible]	[illegible]	[illegible]	-.151	-1.413	.165
X6	[illegible]	[illegible]	111.711	.361	.493	.631

Multiple Linear Regression Model # 5

```
REGRESSION TITLE . . . . . . . . . . . . . . . . . . .SURSIM SURVIVABILITY SIMULATION DATA
DEPENDENT VARIABLE . . . . . . . . . . . . . . . . .    42 A2
TOLERANCE . . . . . . . . . . . . . . . . . . . . .  .0100
ALL DATA CONSIDERED AS A SINGLE GROUP

MULTIPLE R            .9285        STD. ERROR OF EST.      56.4376
MULTIPLE R-SQUARE     .8622

ANALYSIS OF VARIANCE
                  SUM OF SQUARES    DF    MEAN SQUARE     F RATIO   P(TAIL)
     REGRESSION      768963.905     23     33433.213      10.679    .00000
     RESIDUAL        122933.080     40      3073.327
```

VARIABLE		COEFFICIENT	STD. ERROR	STD. REG COEFF	T	P(2 TAIL)	
INTERCEPT		494.399					
I8	4	49.314	30.965	.177	1.592	.119	
I0	5	43.182	21.913	.158	1.971	.056	
I6	6	-29.091	10.633	-.739	-2.736	.009	
I9	9	-19.667	24.162	-.083	-.814	.420	
I10	10	7.471	4.572	.380	1.634	.110	
I12	12	1.214	.590	.360	2.058	.046	
I13	13	-43.278	19.664	-.193	-2.201	.034	
I0	14	29.259	23.374	.104	1.209	.234	
I5	15	77.730	22.127	.285	3.513	.001	
IF	16	67.464	24.799	.247	2.720	.010	
I15	17	.579	.351	.127	1.652	.106	
I17	19	49.416	30.254	.201	1.609	.117	
R2	26						REDUNDANT
R4	28	.003	.010	.019	.276	.784	
S1	31	-456.723	142.406	-.869	-3.207	.003	
S2	32	96.453	82.641	.258	1.167	.250	
S4	34	1.6.012	70.575	.165	1.502	.141	
S6	36	190.294	80.620	.543	2.236	.031	
S8	38	-9.368	6.717	-.346	-1.459	.152	
S9	39	10.541	5.549	.263	1.899	.065	
A4	44	-.007	.003	-.298	-2.576	.015	
A7	47	-5.276	-1.272	-.016	-.152	.840	
X3	50	-.001	.001	-.173	-1.169	.249	
X5	52	-144.244	56.930	-.385	-2.534	.015	

Multiple Linear Regression Model # 6


```
REGRESSION TITLE . . . . . . . . . . . . . . . . . . .SURSIM SURVIVABILITY SIMULATION DATA
DEPENDENT VARIABLE . . . . . . . . . . . . . . . . .    [illegible]
TOLERANCE . . . . . . . . . . . . . . . . . . . . .  .010[illegible]
ALL DATA CONSIDERED AS A SINGLE GROUP

MULTIPLE R            [illegible]        STD. ERROR OF EST.      [illegible]
MULTIPLE R-SQUARE     [illegible]

ANALYSIS OF VARIANCE
                  SUM OF SQUARES    DF    MEAN SQUARE     F RATIO   P(TAIL)
     REGRESSION      [illegible]        [illegible]     3358[illegible]    11.[illegible]    [illegible]
     RESIDUAL        [illegible]        [illegible]     [illegible]
```

VARIABLE		COEFFICIENT	STD. ERROR	STD. REG COEFF	T	P(2 TAIL)	
INTERCEPT		[illegible]					
[illegible]	[illegible]	[illegible]	[illegible]	.176	1.599	[illegible]	
[illegible]	[illegible]	[illegible]	[illegible]	.155	1.862	[illegible]	
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	
[illegible]	[illegible]	[illegible]	[illegible]	.236	3.114	[illegible]	
[illegible]	[illegible]	[illegible]	[illegible]	.351	[illegible]	[illegible]	
[illegible]	[illegible]	[illegible]	[illegible]	.133	[illegible]	[illegible]	
I17	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	
[illegible]							REDUNDANT
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	-1.275	[illegible]	
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	
[illegible]	[illegible]	[illegible]	[illegible]	.333	[illegible]	[illegible]	
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	.174	
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	-.425	[illegible]	
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	
X5	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	

Multiple Linear Regression Model # 7

REGRESSION TITLE . . . . SURVIVABILITY SIMULATION DATA
DEPENDENT VARIABLE. 42 A2
TOLERANCE . . . . [illegible]
ALL DATA CONSIDERED AS A SINGLE GROUP

MULTIPLE R [illegible] STD. ERROR OF EST. [illegible]
MULTIPLE R-SQUARE [illegible]

ANALYSIS OF VARIANCE

	SUM OF SQUARES	DF	MEAN SQUARE	F RATIO	P(TAIL)
REGRESSION	[illegible],142	23	33443.799	[illegible]	[illegible]
RESIDUAL	[illegible]	[illegible]	[illegible]		

VARIABLE		COEFFICIENT	STD. ERROR	STD. REG COEFF	T	P(2 TAIL)	
INTERCEPT		[illegible]					
I5	[illegible]	[illegible]	[illegible]	.175	1.590	.120	
I6	[illegible]	[illegible]	21.862	.159	1.967	.056	
I6	[illegible]	[illegible]	[illegible]	-.728	-2.741	[illegible]	
I7	[illegible]	[illegible]	[illegible]	[illegible]	.444	[illegible]	
I9	[illegible]	[illegible]	[illegible]	[illegible]	-.817	.413	
I10	[illegible]	[illegible]	[illegible]	.377	1.626	.112	
I12	[illegible]	1.213	[illegible]	.361	2.070	.045	
I13	[illegible]	[illegible]	[illegible]	-.193	-2.297	[illegible]	
I[illegible]	[illegible]	[illegible]	23.317	[illegible]	1.199	.238	
I[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	3.526	[illegible]	
I[illegible]	[illegible]	[illegible]	24.741	.247	2.724	[illegible]	
I15	[illegible]	[illegible]	[illegible]	.125	1.646	[illegible]	
I17	[illegible]	[illegible]	[illegible]	[illegible]	1.630	.111	
R2	[illegible]						REDUNDANT
S1	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	
S2	[illegible]	[illegible]	[illegible]	.252	1.139	.262	
S4	[illegible]	[illegible]	[illegible]	[illegible]	1.517	.137	
S5	[illegible]	[illegible]	[illegible]	.543	2.239	.031	
S8	[illegible]	[illegible]	5.701	[illegible]	[illegible]	.156	
S9	[illegible]	[illegible]	5.455	.257	1.874	.068	
A4	[illegible]	[illegible]	[illegible]	-.293	-2.552	.015	
A7	[illegible]	[illegible]	[illegible]	[illegible]	-.158	.875	
X3		[illegible]	[illegible]	-.179	-1.146	[illegible]	
X5	[illegible]	[illegible]	[illegible]	[illegible]	-2.535	[illegible]	

Multiple Linear Regression Model # 8

149

REGRESSION TITLE . . . . SURVIVABILITY SIMULATION DATA
DEPENDENT VARIABLE 42 A2
TOLERANCE . . . . [illegible]
ALL DATA CONSIDERED AS A SINGLE GROUP

MULTIPLE R [illegible] STD. ERROR OF EST. [illegible]
MULTIPLE R-SQUARE [illegible]

ANALYSIS OF VARIANCE

	SUM OF SQUARES	DF	MEAN SQUARE	F RATIO	P(TAIL)
REGRESSION	[illegible]	[illegible]	32134.[illegible]	[illegible]	[illegible]
RESIDUAL	[illegible]	[illegible]	[illegible]		

VARIABLE		COEFFICIENT	STD. ERROR	STD. REG COEFF	T	P(2 TAIL)	
INTERCEPT		[illegible]					
I[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	1.398	[illegible]	
I6	[illegible]	[illegible]	21.367	.133	1.730	[illegible]	
I[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	-2.762	[illegible]	
I[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	
I10	[illegible]	[illegible]	[illegible]	[illegible]	.591	[illegible]	
I12	[illegible]	[illegible]	[illegible]	.232	1.206	[illegible]	
I13	[illegible]	[illegible]	[illegible]	[illegible]	-2.579	[illegible]	
I[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	.618	.54[illegible]	
I[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	
I[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	
I15	[illegible]	[illegible]	[illegible]	[illegible]	1.949	[illegible]	
I17	[illegible]	[illegible]	[illegible]	[illegible]	1.432	[illegible]	
R2	[illegible]						REDUNDANT
R4	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	.857	
S1	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	
S2	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	.303	
S4	[illegible]	[illegible]	[illegible]	[illegible]	1.671	[illegible]	
S5	[illegible]	[illegible]	[illegible]	.225	[illegible]	.275	
S7	[illegible]	[illegible]	[illegible]	[illegible]	.926	.414	
S8	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	.323	
S9	[illegible]	[illegible]	[illegible]	[illegible]	1.526	.112	
A4	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	
A5	[illegible]	[illegible]	[illegible]	[illegible]	1.642	[illegible]	
A6	[illegible]	[illegible]	[illegible]	[illegible]	-3.152	[illegible]	
X[illegible]		[illegible]	[illegible]	-.155	-1.014	.317	

Multiple Linear Regression Model # 9

150

REGRESSION TITLE . . . . COURSIN SURVIVABILITY SIMULATION DATA
DEPENDENT VARIABLE. 42 A2
TOLERANCE 0100
ALL DATA CONSIDERED AS A SINGLE GROUP

MULTIPLE R .9296 STD. ERROR OF EST. [illegible]
MULTIPLE R-SQUARE .864[illegible]

ANALYSIS OF VARIANCE	SUM OF SQUARES	DF	MEAN SQUARE	F RATIO	P(TAIL)
REGRESSION	[illegible]	26	[illegible]	9.666	[illegible]
RESIDUAL	[illegible]	[illegible]	[illegible]		

VARIABLE	COEFFICIENT	STD. ERROR	STD. REG COEFF	T	P(2 TAIL)
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

Multiple Linear Regression Model # 10

ACKNOWLEDGEMENTS

While the inspiration to begin a dissertation must come from within, the motivation to finish comes from family, friends, advisors and sponsors. To all these do I owe deep appreciation for a growing experience.

The author is greatly indebted to her advisor, Professor Richard A. DeMillo, for his interest, insight, guidance and competence and to her doctoral committee: Professor Douglas C. Montgomery, Dr. Charles R. Vick, Professor A. Peter Jensen, and Professor Nancy A. Lynch for the benefit of their constructive comments, suggestions and sense of vision.

A special thanks is given to two friends: Dr. Douglas E. Wrege, whose belief and confidence was always there, and Mr. Clyde G. Roby for his technical talent.

Above all, I am grateful to my husband and our family - to Sam whose love, patience and understanding comforted the tired hours, to my dear son, William, for sensitivity and understanding well beyond his years, and to my sweet daughter, Christine, for her smiles and laughter.

This research was sponsored in part by the U.S. Army Research Office, Research Triangle Park, North Carolina, and the U.S. Army Communications Research and Development Command, Fort Monmouth, New Jersey, under Contract DAAG29-79-C-0118. Gratitude is expressed to these organizations for their essential support.

VITA

Edith W. Martin was born on June 25, 1945 in Chicago, Illinois. She received her Bachelor of Arts degree in psychology from Lake Forest College in 1967 and Master of Science degree in information and computer science from the Georgia Institute of Technology in 1976. She is married to Professor C. Samuel Martin and has two children, William McNutt Martin, III born December 13, 1971 and Christine Katherine Martin born December 23, 1979. She has been a member of the research and management staff of the Engineering Experiment Station at the Georgia Institute of Technology since 1976. Her professional activities include being a member of the editoral review board of Military Electronics Countermeasures, associate editor of the Journal of Systems and Software, chairman of the Computer Architecture Review Subcommittee of the Electronic Industry Association and member of the Institute of Electrical and Electronic Engineers and Association of Computing Machinery.

BIBLIOGRAPHY

1. Baron, P., "On Distributed Networks," IEEE Trans. Comm. Tech. COM-12, pp. 1-9, 1964.

2. Beaudry, M. C., "Performance Related Reliability Measures for Computing Systems, " Proceedings of the Seventh Annual *International Conference on Fault-Tolerant Computing*, Los Angeles, Ca., pp. 16-21, June, 1977.

3. Bell System Technical Journal, Vol. 56, No. 7, September, 1977.

4. Borgerson, B. R. and R. F. Freitas, "A Reliability Model for Gracefully Degrading and Standby-sparing Systems," IEEE Trans. on Computers, Vol. C-24, pp. 517-525, May, 1975.

5. Box, G. E. and Hunter, J. S., "The 2 Fractional Factorial Designs Part I," Technometrics, Vol. 3, No. 3, pp. 311-351, August, 1961.

6. Box, G. E. and Hunter, J. S., "The 2 Fractional Factorial Designs Part II," Technometrics, Vol. 3, No. 4, pp. 449-458, November, 1961.

7. DeMillo, R. A., Lipton, R. J. "Software Project Forcasting," School of Information computer Science, GIT-ICS-80/09 Georgia Institute of Technology, October, 1980.

8. Enslow, P. E., "What is a Distributed Processing System?" Computer, *pp. 13-21, January, 1978.*

9. Foerster, R. E., "Methodology to Evaluate Strategic Command and Control Systems," Technical Report for HQ USAF, Assistant Chief of Staff Studies and Analysis under Contract No. F44620-74-C-0046, July, 1974.

10. Frank, H. and Frish, I. T., "Analysis and Design of [illegible] Networks," IEEE Trans. Comm. Tech., Vol. [illegible], pp. 501-519, October, 1970.

11. Frank, H., "Vulnerability of Communication Networks," [illegible] Trans. Comm. Tech., Vol. COM-15, pp. [illegible]

12. Gay, F. A., "Performance Modeling for [illegible] Systems," Ph.D. Dissertation, Computer [illegible] Northwestern University, June, 1979.

13. Helmer, O., Rescher, N., "On the Epistemology of the [illegible] Sciences," Rand Corporation Report No. [illegible]

14. Hilborn, G., "Measures for distributed processing network survivability," AFIPS Conference Proceedings, National ComputerConference 1980, Vol. 49, pp. 157-163, May, 1980.

15. Jensen, D. E., "The Honeywell Experimental Distributed Processor - An Overview," Computer, pp. 28-37, January, 1978.

16. Losq, J., "A Highly Efficient Redundancy Scheme: Self-Purging Redundancy," IEEE Trans. on Computers, Vol. C-25, pp. 569-578, June, 1976.

17. Losq, J., "Effects of Failure on Gracefully Degradable Systems," Proceedings of the Seventh Annual International Conference on Fault-Tolerant Computing, Los Angeles, Ca., pp. 29-34, June, 1977.

18. Mathur, F. P., Atrzienis, A., "Reliability Analysis and Architecture of a hybrid-redundant digital system: Generalized triple modular redundancy with self-repair," 1970 SJCC, AFIPS Conference Proceedings, Vol. 36, pp. 375-383, 1970.

19. Merwin, R. E. and Mirhakak, M., "Derivation and Use of a Survivability criterion for DDP systems," AFIPS Conference Proceedings, National Computer Conference 1980, Vol. 49, pp. 139-146, May, 1980.

20. Montgomery, D. C., Peck, E. A., Introduction to Regression Analysis, John Wiley & Sons, Inc., 1981.

21. Montgomery, D. C., "Methods for Factor Screening in Computer Simulation experiments," Technical Report Office of Naval Research, Contract N0014-78-C-0312, March, 1979.

22. Ng, Y. and Avizienis, A., "A Reliability Model for Gracefully Degrading and Repairable Fault-Tolerant Systems," Proceedings of the Seventh Annual International Conference on Fault-Tolerant Computing, Los Angeles, Ca., pp. 22-28, June, 1977.

23. Ng, Y., "Reliability Modeling and Analysis for Fault-Tolerant Computers," Ph.D. Dissertation, Computer Science Department, UCLA, UCLA-ENG-7698, September, 1976.

24. Perlis, A., Sayward, S., Shaw. M., Unpublished notes on software metrics, April, 1980.